SBIR-04-09-5050
Release Date 1/6/90

ANCO

1311.05

PROVIDING STRUCTURAL MODULES WITH SELF-INTEGRITY MONITORING SOFTWARE USER'S MANUAL

Prepared for

NASA
JET PROPULSION LABORATORY
Pasadena, California
(NASA SBIR Phase II Contract NAS7-961)

Approval Signatures

Project Mgr./Date

5/4/90
Technical QA/Date Editorial QA/Date

5/4/90
Chief Engineer/Date

Prepared by

The Technical Staff
ANCO ENGINEERS, INC.
9937 Jefferson Boulevard
Culver City, California 90232-3591
(213) 204-5050

April 1990

TABLE OF CONTENTS

LIST OF TABLES

1.0 INTRODUCTION

National Aeronautics and Space Administration (NASA) Contract NAS7-961 [A Small Business Innovation and Research (SBIR) contract from NASA] involved research dealing with remote structural damage detection using the concept of substructures [1]. Several approaches were developed; the main two were: (1) the module (substructure) transfer function matrix (MTFM) approach and (2) modal strain energy distribution method (MSEDM). Either method can be used with a global structure; however, the focus was on substructures.

As part of the research contract, computer software was to be developed which would implement the developed methods. This was done and it was used to process all the finite element generated numerical data for the research. The software was written for the IBM AT personal computer. Copies of it were placed on floppy disks (see Section 4.0).

This report serves as a user's manual for the two sets of damage detection software. Sections 2.0 and 3.0 discuss the use of the MTFM and MSEDM software, respectively.

2.0 MODULE TRANSFER FUNCTION MATRIX METHOD

This method/approach is discussed in detail in Section 3.0 of Reference 1. The corresponding software consists of several programs. To discuss this, it is easiest to describe the process followed in doing the related part of the research.

The basic steps followed are given as follows:

a) obtain transient response of finite element model corresponding to undamaged and damaged states; this was done using COSMOS/M [2];

b) process transient response data to obtain corresponding data in MAC/RAN IV [3] format;

c) run program for determining the module transfer function matrix [MTFM] for the undamaged and damaged states; and

d) determine differences between the MTFM for the two states.

The above steps were followed for each damage level (state) for a given model.

The MTFM software was built around the computer code MAC/RAN IV. This code is a well developed digital time series analysis package. The most important part of it that is related to the subject research is its module (subroutine) TRANS; it is a multi-input, single-output (MISO) code. However, the research required a multi-input, multi-output (MIMO) code. To deal with this, a computer code was written that would execute TRANS for each required output (one at a time)--the code looped through TRANS the number of times needed. The TRANS output files were merged together to form a rectangular matrix corresponding to the MIMO problem.

The details of how to run the MTFM and damage indicator computer codes are described in Table 2.1. All the codes referenced in this table can be found on each of the 5-1/4-in. floppy disks entitled, "Structural Module Integrity Software, ANCO Engineers, Inc., Report 1311.05, August 1988, NASA Contract NAS7-961." Table 2.2 describes the contents of the floppies that is related to the MTFM method. Appendix A contains a listing of the contents of a floppy, including the source listings.

An example of an MTFM and damage indicator run is given in Table 2.3.

TABLE 2.1: RUNNING THE MTFM AND DAMAGE INDICATOR COMPUTER CODES

The codes were written to be used in connection with the finite element computer code COSMOS/M. The sequence followed in running the software package is given as follows:

<u>Step</u> <u>Description</u>

1 Start process using COSMOS/M output time histories corresponding to the undamaged and damaged substructure states.

2 Use macro NASAMAC.BAT - this macro reads the COSMOS/M output time histories and outputs time histories in MAC/RAN IV format.

To do:

2-1 Create NASAMAC.BAT --

```
INTRFS.EXE
MAC423 /R 10000 <FRMCHK.SCF >MACOUT.OUT
DEL MACOUT.OUT
NASADAM
```

2-2 Create NASADAM --

```
NASA.EXE *
NASAD
```

* NASA.EXE creates NASAD.BAT, NASAREN.BAT and MATRX.DAT.

2-3 Run NASAMAC.BAT

Input to INTRFS.EXE is given as follows:

a) Enter COSMOS plot filename to process

b) Enter output MACRAN.SCF filename - include extension .SCF

c) Enter output filename MACRAN.SDF filename - include extension .SDF

d) Enter the number of points in the time history

e) Do you want to truncate the number of points?

f) Enter the number of loads to use from COSMOS

g) Is there load data in the COSMOS plot file?

h) Enter the number of accelerations to use from COSMOS (this must be the total number of node points in the COSMOS output file)

i) Enter channel description for each acceleration used, e.g., -->NODE1,TX; -->NODE6,TX; -->NODE9,TX

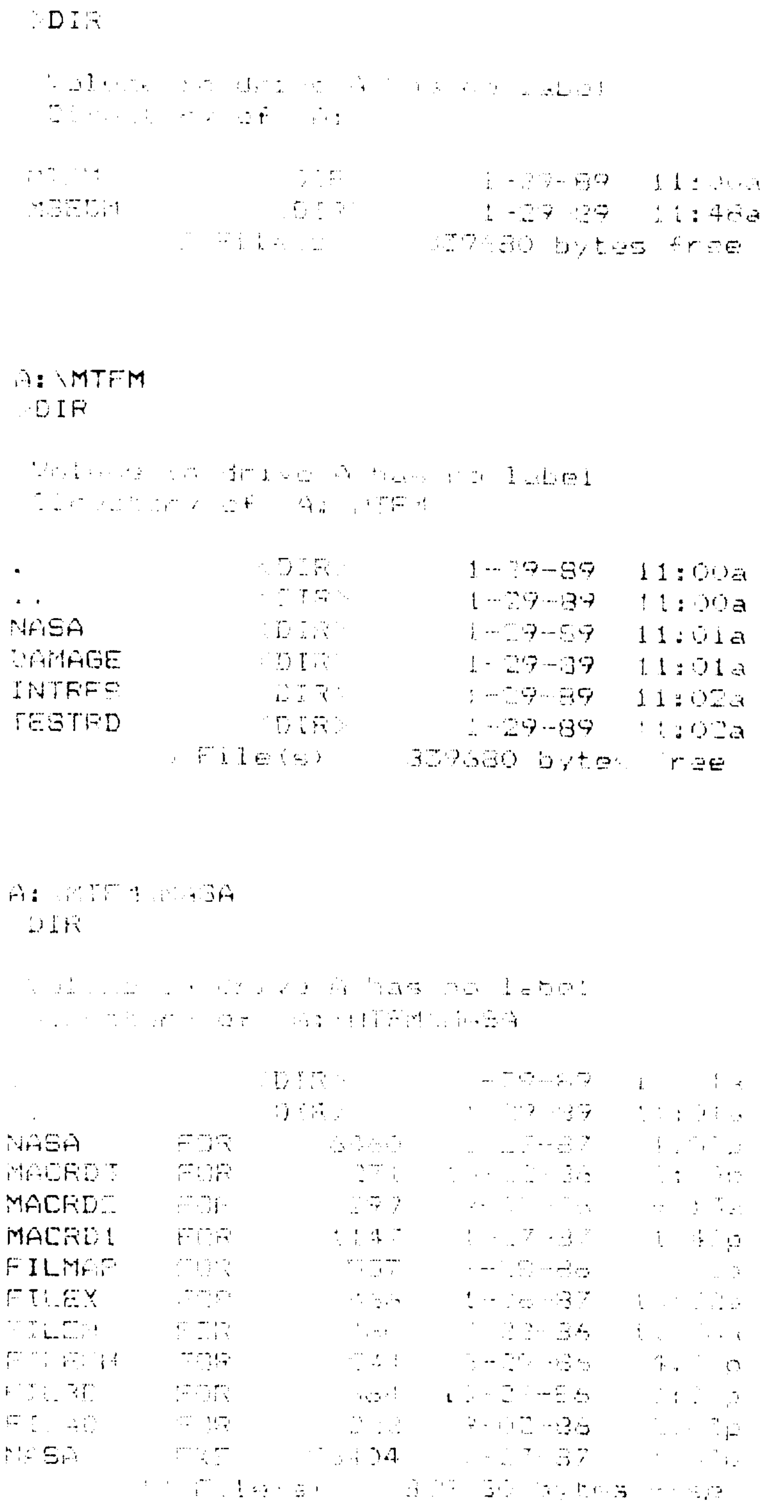
TABLE 2.1 (continued)

Step	Description
3	Input for NASA.EXE -- This program is used to create a batch file which is used to execute MAC/RAN once for each output channel*. Input is given as follows: a) Enter input MACRAN FILENAME - including extension .SDF b) Number of points (time history points) c) Number of points/segments (the number of segments refers to how many segments the time histories are broken up into; for this study four (4) segments were used) d) Enter output filename - no file extensions please e) Enter the number of input channels (<=10) (this is the number of inputs to use for the calculation of the transfer function matrix) f) Specify MAC/RAN channel to use for each input (in MAC/RAN, the first channel is time (values of time in time history); the remaining channels correspond to the node numbers and their order in the COSMOS output file, e.g., time, Node 1, Node 6, and Node 9 correspond to MAC/RAN Channels 1, 2, 3 and 4, respectively. g) Specify MAC/RAN channel for each output h) Is this a (d)amaged or (u)ndamaged run? (Enter "D" or "U") i) Enter a descriptive title for this run ——— * MAC/RAN is a multi-input, single-output transfer function processor code.
4	At this point in the process, a multi-input, multi-output transfer function matrix has been created for the substructure. One matrix is created for each run of INTRFS.EXE, either undamaged or damaged condition. A matrix for both damaged states must be obtained before proceeding with the analysis. The elements of the MTFM are plotted by editing the MTFM data file to obtain the needed data for the element, and then by using the plotting features in MAC/RAN

TABLE 2.1 (continued)

Step	Description
5	Run the program DAMAGE.EXE The final step in this process is to calculate the damage indicators. The results are plotted using the plotting features in MAC/RAN

TABLE 2.2: CONTENTS OF FLOPPY DISK THAT ARE RELATED TO THE MTFM METHOD



```
>DIR

 Volume in drive A has no label
 Directory of  A:\

MTFM          <DIR>       1-29-89   11:00a
MSECM         <DIR>       1-29-89   11:46a
        2 File(s)     327680 bytes free


A:\MTFM
>DIR

 Volume in drive A has no label
 Directory of  A:\MTFM

.             <DIR>       1-29-89   11:00a
..            <DIR>       1-29-89   11:00a
NASA          <DIR>       1-29-89   11:01a
DAMAGE        <DIR>       1-29-89   11:01a
INTRFS        <DIR>       1-29-89   11:02a
TESTRD        <DIR>       1-29-89   11:02a
        6 File(s)     329680 bytes free


A:\MTFM\NASA
>DIR

 Volume in drive A has no label
 Directory of  A:\MTFM\NASA

.             <DIR>       1-29-89   11:01a
..            <DIR>       1-29-89   11:01a
NASA     FOR      6360    1-27-87   [illegible]
MACRDT   FOR       371    1-[illegible]-86   [illegible]
MACRDI   FOR       197    [illegible]   [illegible]
MACRD1   FOR      1147    1-27-87   1:41p
FILMAP   FOR       707    [illegible]-86   [illegible]
FILEX    FOR      [illegible]    1-26-87   [illegible]
FILEM    FOR      [illegible]    [illegible]-22-86   [illegible]
FILRDM   FOR       741    5-29-85   [illegible]
FILRD    FOR      [illegible]    10-21-86   [illegible]
FIL40    FOR       212    9-02-86   [illegible]
NASA     PRT     [illegible]04    1-27-87   [illegible]
       13 File(s)     3[illegible]80 bytes free
```

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 2.2 (continued)

```
A:\MTFM\DAMAGE
>DIR

 Volume in drive A has no label
 Directory of  A:\MTFM\DAMAGE

.            <DIR>      1-29-89  11:01a
..           <DIR>      1-29-89  11:01a
MACWRT   FOR     1775   1-07-87   9:50a
DAMCOM   FOR [illegible]   1-27-87   2:04p
DAMAGE   FOR [illegible]   1-27-87   2:0[illegible]p
DAMAGE   EXE    19876   1-27-87   2:05p
        6 File(s)    309680 bytes free


A:\MTFM\INTRFS
>DIR

 Volume in drive A has no label
 Directory of  A:\MTFM\INTRFS

.            <DIR>      1-29-89  [illegible]
..           <DIR>      1-29-89  [illegible]
OUT1     FOR     1470   1-20-87  [illegible]
INTRFS   FOR     7660   1-07-87  [illegible]a
INTRFS   EXE    10474   1-20-87  [illegible]
        5 File(s)    309680 bytes free


A:\MTFM\TESTRD
>DIR

 Volume in drive A has no label
 Directory of  A:\MTFM\TESTRD

.            <DIR>      [illegible]
..           <DIR>      [illegible]
TESTRD   FOR  [illegible]
TESTRD   EXE  [illegible]
        4 File(s)  [illegible] bytes free
```

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 2.3: EXAMPLE OF MTFM METHOD

The input to this set of programs is rather lengthy, therefore, the input and output listings for the problem discussed in Subsection 4.2.2 of Reference 1 is shown below. The COSMOS/M time thistory plot files used are FRMWWUX.PLT and FRMWWD2X.PLT for the undamaged and damaged states, respectively.

```
INTRFS - PROGRAM TO REFORM COSMOS FILE TO MACRAN FILE
VIPAC-VIBRATION PACKAGE      REVISION NO.:  1.00
ANCO ENGINEERS, INC. (C) Copyright 1984,1985,1986

ENTER COSMOS PLOT FILENAME TO PROCESS : FRMWWUX.PLT

ENTER OUTPUT MACRAN .SCF FILENAME - INCLUDE .SCF : FRMWWUX.SCF

ENTER OUTPUT MACRAN .SDF FILENAME - INCLUDE .SDF : FRMWWUX.SDF

ENTER THE NUMBER OF POINTS IN THE TIME HISTORY : 1024
DO YOU WANT TO TRUNCATE THE NUMBER OF POINTS? N

ENTER THE NUMBER OF LOADS TO USE FROM COSMOS : 0

IS THERE LOAD DATA IN THE COSMOS PLOTFILE? : N**



ENTER THE NUMBER OF ACCELERATIONS TO USE FROM COSMOS : 3

ENTER CHANNEL DESCRIPTION FOR EACH ACCELERATION USED
--> NODE1.TX
--> NODE6,TX
--> NODE9,TX

  FILE HAS A TIME STEP OF :  2.000E-02 SECONDS

 READING IN ACCELERATION DATA

 REFORMATTING TO COSMOS -.SCF FORMAT


Stop - Program terminated.

21:14:40
E:\
>
```

TABLE 2.3 (continued)

```
21:14:40
E:\
>TYPE FRMWWUX.PLT
    1024   9 time d 1     1 v 1     1 a 1     1 d 1     6 v 1     6 a 1     6 d 1
 1     9 a 1
 .20000E-01  .0000      .0000       .0000       .0000       .0000       .0000
0000      .0000      .0000
 .40000E-01  .0000      .0000       .0000       .0000       .0000       .0000
0000      .0000      .0000
 .60000E-01  .0000      .0000       .0000       .0000       .0000       .0000
0000      .0000      .0000
 .80000E-01  .0000      .0000       .0000       .0000       .0000       .0000
0000      .0000      .0000
 .10000E+00  .0000      .0000       .0000       .0000       .0000       .0000
0000      .0000      .0000
 .12000      .0000      .0000       .0000       .0000       .0000       .0000
0000      .0000      .0000
 .14000      .0000      .0000       .0000       .0000       .0000       .0000
0000      .0000      .0000
 .16000      .0000      .0000       .0000       .0000       .0000       .0000
0000      .0000      .0000
 .18000      .0000      .0000       .0000       .0000       .0000       .0000
0000      .0000      .0000
 .20000      .0000

 1.9800      .0000      .0000       .0000       .0000       .0000       .0000
0000      .0000      .0000
 2.0000      .0000      .0000       .0000       .0000       .0000       .0000
0000      .0000      .0000
 2.0200      .0000      .0000       .2919    -.2034E-05 -.2034E-03   .2716
3388E-04 -.3388E-02 -.4687E-01
 2.0400      .0000      .0000       .9292    -.1372E-04 -.9651E-03   .8734
2285E-03 -.1607E-01 -.5564E-03
 2.0600      .0000      .0000       1.891    -.4776E-04 -.2439E-02  1.799
7955E-03 -.4063E-01  .3654
 2.0800      .0000      .0000       2.283    -.1111E-03 -.3890E-02  2.230
1850E-02 -.6479E-01  1.392
 2.1000      .0000      .0000       .5753    -.1798E-03 -.2979E-02   .7200
2994E-02 -.4961E-01  2.985
 2.1200      .0000      .0000      -1.930    -.1919E-03  .1759E-02 -1.600
3197E-02  .2930E-01  3.552
 2.1400      .0000      .0000      -2.245    -.1006E-03  .7372E-02 -2.013
1676E-02  .1228      1.621
 2.1600      .0000      .0000      -.3218     .6125E-04  .8818E-02 -.4093
1020E-02  .1468     -1.779
 2.1800      .0000      .0000       .5932     .2015E-03  .5206E-02   .3196
3356E-02  .8670E-01 -3.964
 2.2000      .0000      .0000      -.4468     .2570E-03  .3414E-03 -.6596
4280E-02  .5685E-02 -3.991
 2.2200      .0000      .0000
```

TABLE 2.3 (continued)

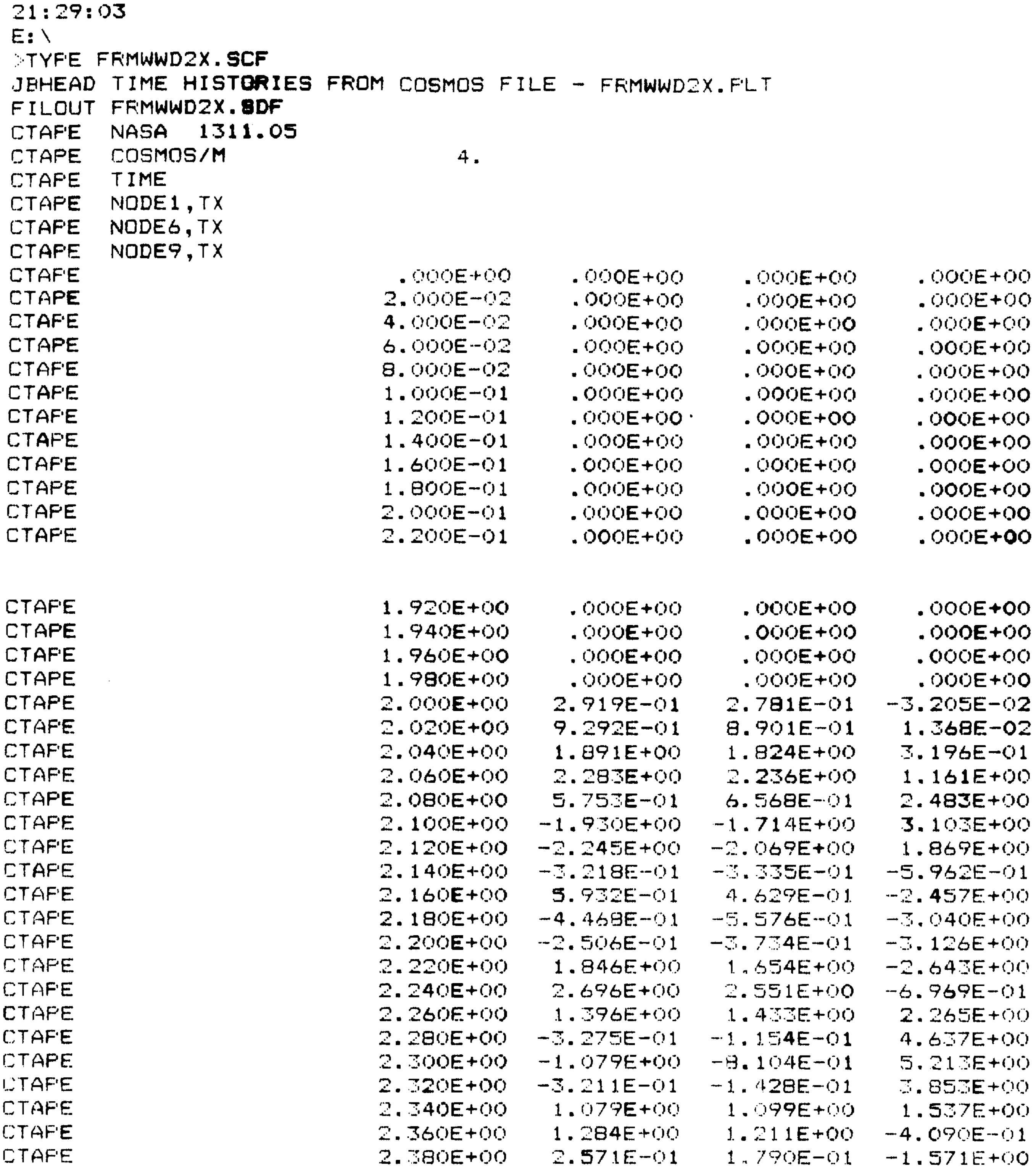
```
21:16:42
E:\
>TYPE FRMWWUX.SCF
JBHEAD TIME HISTORIES FROM COSMOS FILE - FRMWWUX.FLT
FILOUT FRMWWUX.SDF
CTAPE  NASA  1311.05
CTAPE  COSMOS/M            4.
CTAPE  TIME
CTAPE  NODE1,TX
CTAPE  NODE6,TX
CTAPE  NODE9,TX
CTAPE                       .000E+00    .000E+00    .000E+00    .000E+00
CTAPE                      2.000E-02    .000E+00    .000E+00    .000E+00
CTAPE                      4.000E-02    .000E+00    .000E+00    .000E+00
CTAPE                      6.000E-02    .000E+00    .000E+00    .000E+00
CTAPE                      8.000E-02    .000E+00    .000E+00    .000E+00
CTAPE                      1.000E-01    .000E+00    .000E+00    .000E+00
CTAPE                      1.200E-01    .000E+00    .000E+00    .000E+00
CTAPE                      1.400E-01    .000E+00    .000E+00    .000E+00
CTAPE                      1.600E-01    .000E+00    .000E+00    .000E+00
CTAPE                      1.800E-01    .000E+00    .000E+00    .000E+00
CTAPE                      2.000E-01    .000E+00    .000E+00    .000E+00
CTAPE                      2.200E-01    .000E+00    .000E+00    .000E+00
CTAPE                      2.400E-01

CTAPE                      1.900E+00    .000E+00    .000E+00    .000E+00
CTAPE                      1.920E+00    .000E+00    .000E+00    .000E+00
CTAPE                      1.940E+00    .000E+00    .000E+00    .000E+00
CTAPE                      1.960E+00    .000E+00    .000E+00    .000E+00
CTAPE                      1.980E+00    .000E+00    .000E+00    .000E+00
CTAPE                      2.000E+00   2.919E-01   2.716E-01  -4.687E-02
CTAPE                      2.020E+00   9.292E-01   8.734E-01  -5.564E-04
CTAPE                      2.040E+00   1.891E+00   1.799E+00   3.654E-01
CTAPE                      2.060E+00   2.283E+00   2.230E+00   1.392E+00
CTAPE                      2.080E+00   5.753E-01   7.200E-01   2.985E+00
CTAPE                      2.100E+00  -1.930E+00  -1.600E+00   3.552E+00
CTAPE                      2.120E+00  -2.245E+00  -2.013E+00   1.621E+00
CTAPE                      2.140E+00  -3.218E-01  -4.093E-01  -1.779E+00
CTAPE                      2.160E+00   5.932E-01   3.196E-01  -3.964E+00
CTAPE                      2.180E+00  -4.468E-01  -6.596E-01  -3.991E+00
CTAPE                      2.200E+00  -2.506E-01  -4.247E-01  -3.150E+00
CTAPE                      2.220E+00   1.846E+00   1.631E+00  -1.721E+00
CTAPE                      2.240E+00   2.696E+00   2.603E+00   1.163E+00
CTAPE                      2.260E+00   1.396E+00   1.594E+00   4.682E+00
CTAPE                      2.280E+00  -3.275E-01   9.231E-02   6.664E+00
CTAPE                      2.300E+00  -1.079E+00  -6.677E-01   5.774E+00
CTAPE                      2.320E+00  -3.211E-01  -1.588E-01   2.383E+00
CTAPE                      2.340E+00   1.079E+00   9.207E-01  -1.565E+00
CTAPE                      2.360E+00   1.284E+00   9.743E-01  -3.870E+00
CTAPE                      2.380E+00   2.571E-01   1
```

TABLE 2.3 (continued)

```
INTRFS - PROGRAM TO REFORM COSMOS FILE TO MACRAN FILE
VIPAC-VIBRATION PACKAGE      REVISION NO.:  1.00
ANCO ENGINEERS, INC. (C) Copyright 1984,1985,1986

ENTER COSMOS PLOT FILENAME TO PROCESS : FRMWWD2X.PLT

ENTER OUTPUT MACRAN .SCF FILENAME - INCLUDE .SCF : FRMWWD2X.SCF

ENTER OUTPUT MACRAN .SDF FILENAME - INCLUDE .SDF : FRMWWD2X.SDF

ENTER THE NUMBER OF POINTS IN THE TIME HISTORY : 1024
DO YOU WANT TO TRUNCATE THE NUMBER OF POINTS? N

ENTER THE NUMBER OF LOADS TO USE FROM COSMOS : 0

IS THERE LOAD DATA IN THE COSMOS PLOTFILE? : N**









ENTER THE NUMBER OF ACCELERATIONS TO USE FROM COSMOS : 3

ENTER CHANNEL DESCRIPTION FOR EACH ACCELERATION USED
--> NODE1,TX
--> NODE6,TX
--> NODE9,TX

 FILE HAS A TIME STEP OF :  2.000E-02 SECONDS

 READING IN ACCELERATION DATA

 REFORMATTING TO COSMOS -.SCF FORMAT


Stop - Program terminated.

21:27:32
E:\
>
```

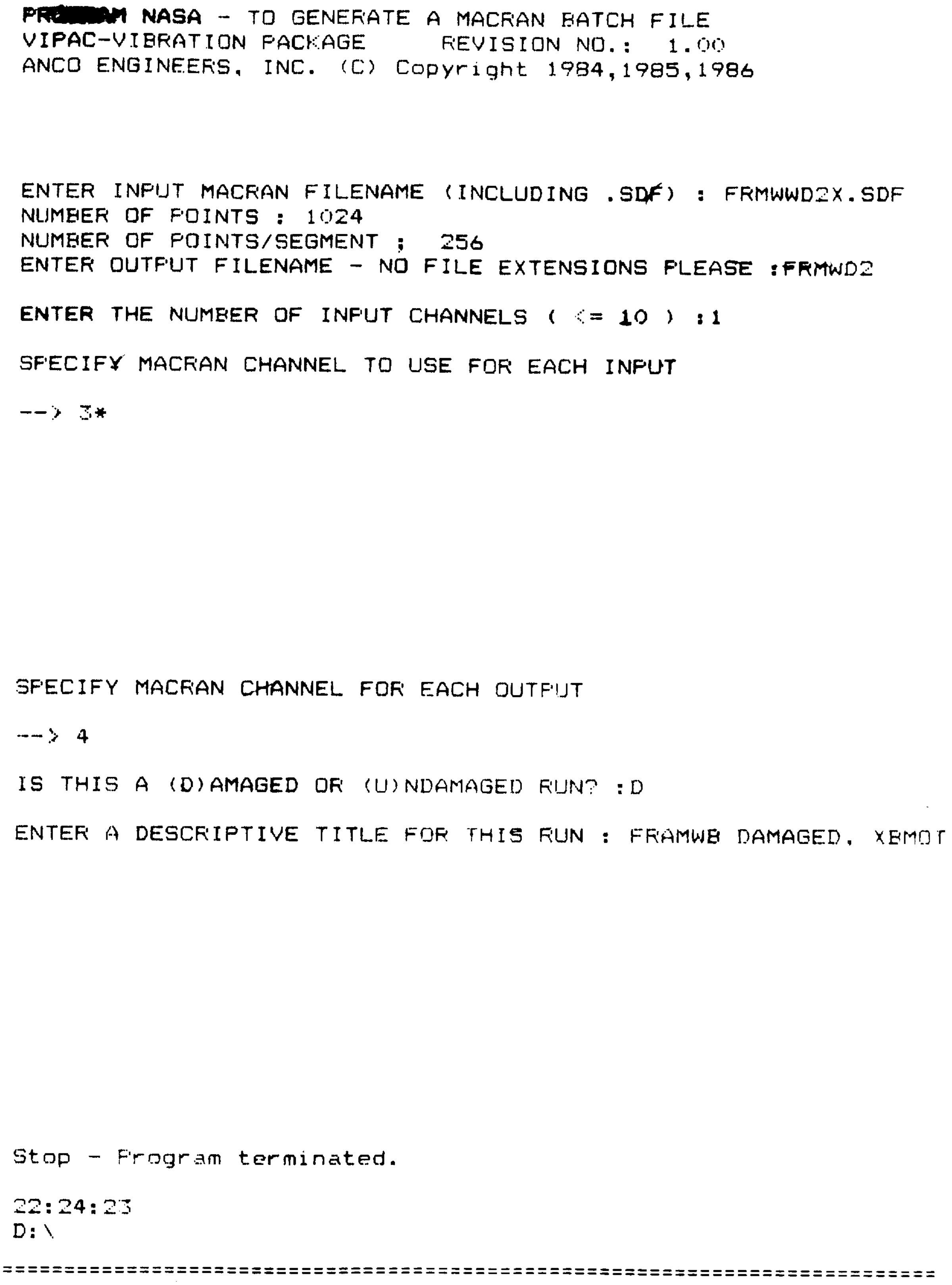
```
E:\
>TYPE FRMWWD2X.PLT
   1024   9 time d 1     1 v 1      1 a 1      1 d 1      6 v 1      6 a 1      6 d 1
 1     9 a 1     9
 .20000E-01  .0000       .0000       .0000       .0000       .0000       .0000
0000       .0000       .0000
 .40000E-01  .0000       .0000       .0000       .0000       .0000       .0000
0000       .0000       .0000
 .60000E-01  .0000       .0000       .0000       .0000       .0000       .0000
0000       .0000       .0000
 .80000E-01  .0000       .0000       .0000       .0000       .0000       .0000
0000       .0000       .0000
 .10000E+00  .0000       .0000       .0000       .0000       .0000       .0000
0000       .0000       .0000
 .12000      .0000       .0000       .0000       .0000       .0000       .0000
0000       .0000       .0000
 .14000      .0000       .0000       .0000       .0000       .0000       .0000
0000       .0000       .0000
 .16000      .0000       .0000       .0000       .0000       .0000       .0000
0000       .0000       .0000
 .18000      .0000       .0000       .0000       .0000       .0000       .0000
0000       .0000       .0000
 .20000      .0000       .0000       .0000       .0000       .0000       .0000
0000       .0000       .0000
 .2

 1.9800      .0000       .0000       .0000       .0000       .0000       .0000
0000       .0000       .0000
 2.0000      .0000       .0000       .0000       .0000       .0000       .0000
0000       .0000       .0000
 2.0200      .0000       .0000       .2919     -.1384E-05 -.1384E-03  .2781
3239E-04 -.3239E-02 -.3205E-01
 2.0400      .0000       .0000       .9292     -.9448E-05 -.6680E-03  .8901
2211E-03 -.1563E-01  .1368E-01
 2.0600      .0000       .0000      1.891      -.3343E-04 -.1730E-02  1.824
7825E-03 -.4050E-01  .3196
 2.0800      .0000       .0000      2.283      -.7955E-04 -.2881E-02  2.236
1862E-02 -.6744E-01  1.161
 2.1000      .0000       .0000       .5753     -.1338E-03 -.2546E-02  .6568
3132E-02 -.5959E-01  2.483
 2.1200      .0000       .0000     -1.930      -.1551E-03  .4194E-03 -1.714
3630E-02  .9818E-02  3.103
 2.1400      .0000       .0000     -2.245      -.1076E-03  .4327E-02 -2.069
2519E-02  .1013      1.869
 2.1600      .0000       .0000      -.3218     -.4676E-05  .5967E-02 -.3335
1094E-03  .1397      -.5962
 2.1800      .0000       .0000       .5932      .1005E-03  .4547E-02  .4629
2351E-02  .1064     -2.457
 2.2000      .0000       .0000      -.4468      .1673E-03  .2136E-02 -.5576
3916E-02  .4999E-01 -3.040
 2.2200      .0000       .0000      -.2506      .1866E-03 -.2003E-03 -.3734
```

TABLE 2.3 (continued)



```
21:29:03
E:\
>TYPE FRMWWD2X.SCF
JBHEAD TIME HISTORIES FROM COSMOS FILE - FRMWWD2X.PLT
FILOUT FRMWWD2X.SDF
CTAPE  NASA  1311.05
CTAPE  COSMOS/M                 4.
CTAPE  TIME
CTAPE  NODE1,TX
CTAPE  NODE6,TX
CTAPE  NODE9,TX
CTAPE                          .000E+00    .000E+00    .000E+00    .000E+00
CTAPE                         2.000E-02    .000E+00    .000E+00    .000E+00
CTAPE                         4.000E-02    .000E+00    .000E+00    .000E+00
CTAPE                         6.000E-02    .000E+00    .000E+00    .000E+00
CTAPE                         8.000E-02    .000E+00    .000E+00    .000E+00
CTAPE                         1.000E-01    .000E+00    .000E+00    .000E+00
CTAPE                         1.200E-01    .000E+00    .000E+00    .000E+00
CTAPE                         1.400E-01    .000E+00    .000E+00    .000E+00
CTAPE                         1.600E-01    .000E+00    .000E+00    .000E+00
CTAPE                         1.800E-01    .000E+00    .000E+00    .000E+00
CTAPE                         2.000E-01    .000E+00    .000E+00    .000E+00
CTAPE                         2.200E-01    .000E+00    .000E+00    .000E+00


CTAPE                         1.920E+00    .000E+00    .000E+00    .000E+00
CTAPE                         1.940E+00    .000E+00    .000E+00    .000E+00
CTAPE                         1.960E+00    .000E+00    .000E+00    .000E+00
CTAPE                         1.980E+00    .000E+00    .000E+00    .000E+00
CTAPE                         2.000E+00   2.919E-01   2.781E-01  -3.205E-02
CTAPE                         2.020E+00   9.292E-01   8.901E-01   1.368E-02
CTAPE                         2.040E+00   1.891E+00   1.824E+00   3.196E-01
CTAPE                         2.060E+00   2.283E+00   2.236E+00   1.161E+00
CTAPE                         2.080E+00   5.753E-01   6.568E-01   2.483E+00
CTAPE                         2.100E+00  -1.930E+00  -1.714E+00   3.103E+00
CTAPE                         2.120E+00  -2.245E+00  -2.069E+00   1.869E+00
CTAPE                         2.140E+00  -3.218E-01  -3.335E-01  -5.962E-01
CTAPE                         2.160E+00   5.932E-01   4.629E-01  -2.457E+00
CTAPE                         2.180E+00  -4.468E-01  -5.576E-01  -3.040E+00
CTAPE                         2.200E+00  -2.506E-01  -3.734E-01  -3.126E+00
CTAPE                         2.220E+00   1.846E+00   1.654E+00  -2.643E+00
CTAPE                         2.240E+00   2.696E+00   2.551E+00  -6.969E-01
CTAPE                         2.260E+00   1.396E+00   1.433E+00   2.265E+00
CTAPE                         2.280E+00  -3.275E-01  -1.154E-01   4.637E+00
CTAPE                         2.300E+00  -1.079E+00  -8.104E-01   5.213E+00
CTAPE                         2.320E+00  -3.211E-01  -1.428E-01   3.853E+00
CTAPE                         2.340E+00   1.079E+00   1.099E+00   1.537E+00
CTAPE                         2.360E+00   1.284E+00   1.211E+00  -4.090E-01
CTAPE                         2.380E+00   2.571E-01   1.790E-01  -1.571E+00
```

TABLE 2.3 (continued)



```
PR[redacted]M NASA - TO GENERATE A MACRAN BATCH FILE
VIPAC-VIBRATION PACKAGE      REVISION NO.:  1.00
ANCO ENGINEERS, INC. (C) Copyright 1984,1985,1986



ENTER INPUT MACRAN FILENAME (INCLUDING .SDF) : FRMWWD2X.SDF
NUMBER OF POINTS : 1024
NUMBER OF POINTS/SEGMENT ;  256
ENTER OUTPUT FILENAME - NO FILE EXTENSIONS PLEASE :FRMWD2

ENTER THE NUMBER OF INPUT CHANNELS ( <= 10 ) :1

SPECIFY MACRAN CHANNEL TO USE FOR EACH INPUT

--> 3*







SPECIFY MACRAN CHANNEL FOR EACH OUTPUT

--> 4

IS THIS A (D)AMAGED OR (U)NDAMAGED RUN? :D

ENTER A DESCRIPTIVE TITLE FOR THIS RUN : FRAMWB DAMAGED, XBMOT








Stop - Program terminated.

22:24:23
D:\
===========================================================================
```

3.0 MODAL STRAIN ENERGY DISTRIBUTION METHOD (MSEDM)

This method (modal strain energy distribution method) is discussed in detail in Section 6.0 of Reference 1. The corresponding software consists of several programs. The basic steps followed in using it are given as follows:

a) run a program to determine the strain energy in each finite element for each mode;

b) determine the level of damage associated with each mode; this can be done by using the damage indicator discussed in Reference 1; and

c) run program to determine possible locations of damage.

The details of how to run the MSEDM computer codes are described in Table 3.1. All of the codes referenced in this table can be found on each of the 5-1/4-in. floppy disks entitled, "Structural Module Integrity Software, ANCO Engineers, Inc., Report 1311.05, August 1988, NASA Contract NAS7-961." Table 3.2 describes the contents of the floppies that are related to the MSEDM. Appendix A contains related source code listings.

An example of an MSEDM run is given in Table 3.3.

TABLE 3.1: RUNNING THE MSEDM COMPUTER CODES

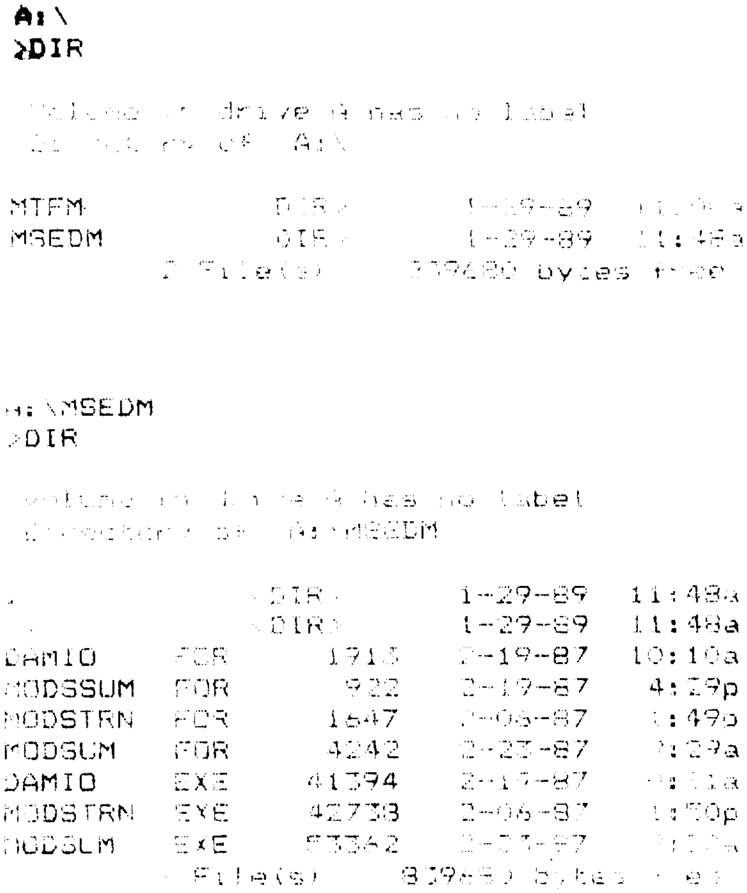
The following three computer codes must be run in using the MSEDM. The required inputs are listed for each code.

MODSTRN - This program computes element modal strain energies for a simply-supported beam due to bending deformation only. The user must input the number of elements and modes to use. The output element modal strain energies are contained in the file "MS.DD". This file is a binary file.

DAMIO - This program allows the user to input damage parameters to be used in damage assessment. Damage parameters are input on a mode-by-mode basis. The user also specifies the output file containing the damage parameters. This output file is an ASCII file.

MODSUM - This program performs the damage assessment based on the modal strain energies and the damage parameters. The user must input the file name containing the modal strain energies and the damage parameters. The user also specifies the damage fraction (g) used in assessing damage. Alternately, the user can select the option of summarizing the element strain energies for a given mode. In either case, output data is contained in the file "MODSUM.DD".

TABLE 3.2: CONTENTS OF FLOPPY DISK THAT ARE RELATED TO THE MSEDM



```
A:\
>DIR

 Volume in drive A has no label
 Directory of  A:\

MTFM          <DIR>        1-29-89  11:48a
MSEDM         <DIR>        1-29-89  11:48a
        2 File(s)     339480 bytes free


A:\MSEDM
>DIR

 Volume in drive A has no label
 Directory of  A:\MSEDM

.             <DIR>        1-29-89  11:48a
..            <DIR>        1-29-89  11:48a
DAMIO    FOR      1713     2-19-87  10:10a
MODSSUM  FOR       922     2-19-87   4:29p
MODSTRN  FOR      1647     2-06-87   1:49p
MODSUM   FOR      4242     2-23-87   [illegible]:29a
DAMIO    EXE     41394     2-17-87   [illegible]:11a
MODSTRN  EXE     42738     2-06-87   1:50p
MODSUM   EXE     [illegible]3362     2-2[illegible]-87   [illegible]
       [illegible] File(s)     839[illegible] bytes free
```

ORIGINAL PAGE IS
OF POOR QUALITY

TABLE 3.3: EXAMPLE OF MSEDM

Computer Code	Description	Input Value
MODSTRN	Damage file used.	DAMOUT3.DD
	Problem title (= damage at N = 0.25 and N = 0.5).	
	Number of modes to use.	12
	Number of elements for simply supported beam	60
	Damage fraction.	0.4
DAMIO	Damage file name.	DAMOUT3.DD
	Problem title (= damage at N = 0.25 and N = 0.5)	
	Number of modes to use.	12
	Modal damage indicator values [scalars $D(\omega_i)$], and corresponding critical values are listed below.	
MODSUM	Damage file used.	DAMOUT3.DD
	Problem title (= damage at N = 0.25 and N = 0.5)	
	Number of modes to use.	12
	Number of elements.	60
	Damage fraction.	0.4

The output from the various computer codes are shown on the following pages:

TABLE 3.3 (continued)

```
SIMPLY SUPPORTED BEAM
DAMAGE FILE USED : DAMOUT3.DD
DAMAGE TITLE : DAMAGE AT N = 0.25 AND AT N = 0.5
NUMBER OF MODES : 12
NUMBER OF ELEMENTS : 60
DAMAGE FRACTION :   .4000

MODE     MAXIMUM ELEMENT    ELEMENT
 NO.      STRAIN ENERGY        NO.

  1         1.000E+00          31
  2         1.000E+00          45
  3         1.000E+00          31
  4         1.000E+00          38
  5         1.000E+00          31
  6         1.000E+00          45
  7         1.000E+00          22
  8         1.000E+00          34
  9         1.000E+00          17
 10         1.000E+00          45
 11         1.000E+00          25
 12         1.000E+00          38


  DAMAGE FILENAME : DAMOUT3.DD
  DAMAGE AT N = 0.25 AND AT N = 0.5
  NUMBER OF MODES : 12
  MODE       D(W)            Dcrit(W)
    1      100.000           10.000
    2      100.000           10.000
    3      100.000           10.000
    4        1.000           10.000
    5      100.000           10.000
    6      100.000           10.000
    7      100.000           10.000
    8        1.000           10.000
    9      100.000           10.000
   10      100.000           10.000
   11      100.000           10.000
   12        1.000           10.000
```

TABLE 3.3 (continued)

```
SIMPLY SUPPORTED BEAM
DAMAGE FILE USED : DAMOUT3.DD
DAMAGE TITLE : DAMAGE AT N = 0.25 AND AT N = 0.5
NUMBER OF MODES : 12
NUMBER OF ELEMENTS : 60
DAMAGE FRACTION :  .4000
```

ELEMENT DAMAGE SUMMARY

ELEMENT NO.	MODE NO.												POSSIBLE DAMAGE
	1	2	3	4	5	6	7	8	9	10	11	12	
1	x	x	x	x	x	x	x	x	x	x	x	x	YES
2	x	x	x	x	x	x	x	x	x	x	x		NO
3	x	x	x	x	x	x	x		x	x	x		NO
4	x	x	x		x	x	x		x	x	x		NO
5	x	x	x		x	x	x		x	x	x	x	NO
6	x	x	x		x	x	x		x	x	x	x	NO
7	x	x	x		x	x	x	x	x	x	x		NO
8	x	x	x		x	x	x	x	x	x	x		NO
9	x	x	x		x	x	x	x	x	x	x		NO
10	x	x	x		x	x	x		x	x	x	x	NO
11	x	x	x		x	x	x		x	x	x	x	NO
12	x	x	x		x	x	x		x	x	x		NO
13	x	x	x	x	x	x	x		x	x	x		NO
14	x	x	x	x	x	x	x	x	x	x	x		NO
15	x	x	x	x	x	x	x	x	x	x	x	x	YES
16	x	x	x	x	x	x	x	x	x	x	x	x	YES
17	x	x	x	x	x	x	x	x	x	x	x		NO
18	x	x	x	x	x	x	x		x	x	x		NO
19	x	x	x		x	x	x		x	x	x		NO
20	x	x	x		x	x	x		x	x	x	x	NO
21	x	x	x		x	x	x		x	x	x	x	NO
22	x	x	x		x	x	x	x	x	x	x		NO
23	x	x	x		x	x	x	x	x	x	x		NO
24	x	x	x		x	x	x	x	x	x	x		NO
25	x	x	x		x	x	x		x	x	x	x	NO
26	x	x	x		x	x	x		x	x	x	x	NO
27	x	x	x		x	x	x		x	x	x		NO
28	x	x	x	x	x	x	x		x	x	x		NO
29	x	x	x	x	x	x	x	x	x	x	x		NO
30	x	x	x	x	x	x	x	x	x	x	x	x	YES
31	x	x	x	x	x	x	x	x	x	x	x	x	YES
32	x	x	x	x	x	x	x	x	x	x	x		NO
33	x	x	x	x	x	x	x		x	x	x		NO
34	x	x	x		x	x	x		x	x	x		NO
35	x	x	x		x	x	x		x	x	x	x	NO
36	x	x	x		x	x	x		x	x	x	x	NO
37	x	x	x		x	x	x	x	x	x	x		NO
38	x	x	x		x	x	x	x	x	x	x		NO
39	x	x	x		x	x	x	x	x	x	x		NO
40	x	x	x		x	x	x		x	x	x	x	NO

x indicates possible modal element damage

TABLE 3.3 (continued)

```
SIMPLY SUPPORTED BEAM
DAMAGE FILE USED : DAMOUT3.DD
DAMAGE TITLE : DAMAGE AT N = 0.25 AND AT N = 0.5
NUMBER OF MODES : 12
NUMBER OF ELEMENTS : 60
DAMAGE FRACTION :   .4000

                  ELEMENT DAMAGE SUMMARY

ELEMENT                        MODE                              POSSIBLE
  NO.                           NO.                               DAMAGE
        1    2    3    4    5    6    7    8    9    10   11   12
  41    x    x    x         x    x    x         x    x    x    x        NO
  42    x    x    x         x    x    x         x    x    x             NO
  43    x    x    x    x    x    x    x         x    x    x             NO
  44    x    x    x    x    x    x    x    x    x    x    x             NO
  45    x    x    x    x    x    x    x    x    x    x    x    x       YES
  46    x    x    x    x    x    x    x    x    x    x    x    x       YES
  47    x    x    x    x    x    x    x    x    x    x    x             NO
  48    x    x    x    x    x    x    x         x    x    x             NO
  49    x    x    x         x    x    x         x    x    x             NO
  50    x    x    x         x    x    x         x    x    x    x        NO
  51    x    x    x         x    x    x         x    x    x    x        NO
  52    x    x    x         x    x    x    x    x    x    x             NO
  53    x    x    x         x    x    x    x    x    x    x             NO
  54    x    x    x         x    x    x    x    x    x    x             NO
  55    x    x    x         x    x    x         x    x    x    x        NO
  56    x    x    x         x    x    x         x    x    x    x        NO
  57    x    x    x         x    x    x         x    x    x             NO
  58    x    x    x    x    x    x    x         x    x    x             NO
  59    x    x    x    x    x    x    x    x    x    x    x             NO
  60    x    x    x    x    x    x    x    x    x    x    x    x.      YES

          x indicates possible modal element damage

=========================================================================
```

4.0 CONTENTS OF FLOPPY DISKS

A single floppy disk contains all the software necessary to solve either the module transfer function matrix*/damage assessment or modal strain energy distribution method problem defined in Reference 1. The software contained on the floppy is listed in Tables 2.2 and 3.2. Also, each floppy contains the input data for the two sample problems described earlier. The software on the floppies consists of the Fortran source listings and the corresponding executable codes.

The floppies are 5-1/4-in., 1.2 MByte, high density 96 TPI. Three floppy disks, with software, have been provided with this report.

* This is true provided that MAC/RAN IV has been installed on the microcomputer being used for these analyses.

5.0 REFERENCES

1. Providing Structural Modules With Self-Integrity Monitoring, Final Report - Phase II; ANCO Engineers, Inc., Report 1311.05, August 1988; prepared for NASA, Jet Propulsion Laboratory, Pasadena, California (NASA SBIR Phase II Contract NAS7-961).

2. COSMOS/M, general purpose structural linear and nonlinear, static and dynamic finite element computer code; Structural and Analysis Corporation, Inc., Santa Monica, California.

3. MAC/RAN IV, Time Series Data Analysis System, University Software Systems, El Segundo, California, 1973.

APPENDIX A

SOURCE CODE LISTINGS FOR
MTFM AND MSEDM APPROACHES

Contained in this appendix are the Fortran source code listings for the two damage assessment methods.

MODULE TRANSFER FUNCTION MATRIX (MTFM)
SOURCE CODE LISTINGS

```
A:\MITFM\NA [illegible]
 DIR

 Volume in drive A has no label
 Directory of  A:\MITFM\NASA


NASA        [illegible]      5480    1-27-87     1:37p
MACRO1      [illegible]       271   10-12-86     3:[illegible]7p
MACROT      [illegible]       159    9-[illegible]-86     4:17a
MACRO1      [illegible]      1147    1-17-87     1:47p
FILMAP      [illegible]  [illegible]    1-10-86     3:21p
FILEX       [illegible]       456    1-06-87    10:22a
FILE[illegible]     [illegible]       540    2-28-86    11:56a
FILE[illegible]H     [illegible]       744    1-29-86     4:37p
FILE3C      [illegible]       654   10-27-86     3:07p
FILE4D      [illegible]       212    9-02-86     1:[illegible]3p
```

NASA.FOR

```
C
C PROGRAM NASA - PURPOSE : TO CREATE A BATCH FILE
C       WHICH WILL EXECUTE MACRAN SEVERAL TIMES
C
          IMPLICIT INTEGER*2 (I-N)
          DIMENSION ICHNI(10),ICHNO(14)
          CHARACTER*80 KEY
          CHARACTER*30 TITLE
          CHARACTER*12 MFIL1I,MFIL1O,MFIL2I,MFIL2O
          CHARACTER*12 MFIL3I,MFIL3O,MFIL4I,MFIL4O
          CHARACTER*12 FILIN,FILOT1,FILOT2
          CHARACTER*6 MFIL1,MFIL2,MFIL3,MFIL4,FILOT
          CHARACTER*3 FX1,FX2,FX3,FX4,AYN
          CHARACTER*1 ATYPE,FILET(12)
          LOGICAL*2 EXISTS,IER
          COMMON/NASADT/ICHNI,ICHNO,NINPUT,NOUTPT,FILIN,NPNTS
          EQUIVALENCE (FILOT2,FILET(1))
C
C
          CALL VCLEAR
          CALL VCURXY(0,0)
C
C DISPLAY PROGRAM NAME TO USER
C
          KEY = 'PROGRAM NASA - TO GENERATE A MACRAN BATCH FILE$'
          CALL PRMPT(KEY,1.00)
C
C BEGNIN HERE
C
C
C OPEN UP FILE CONTAINING NPTS
C
          OPEN(31,FILE='INTRFS.DAT')
          READ(31,8831) NPTS
 8831  FORMAT(I4)
          NPNTS = (NPTS+3)/4
          IR = 7
 10       CONTINUE
          KEY = 'ENTER INPUT MACRAN FILENAME (INCLUDING .SDF) : $'
          CALL KEYBD(KEY,R,I,FILIN,3,(12),0,IR)
C
C SEE IF FILE EXISTS
C
          INQUIRE(FILE=FILIN,EXIST=EXISTS)
          IF(EXISTS) GO TO 15
          IR = IR + 1
          CALL VCURXY(0,IR)
          WRITE(*,9010)
 9010  FORMAT(1X,'FILE DOES NOT EXISTS - PROGRAM ABORT!!')
          STOP
C
C MAKE SURE IT HAS A .SDF EXTENSION
C
 15       CONTINUE
          IR = IR + 1
          CALL VCURXY(0,IR)
          WRITE(*,8543) NPTS,NPNTS
 8543  FORMAT(1X,'NUMBER OF POINTS : ',I4,
     1  /1X,'NUMBER OF POINTS/SEGMENT ; ',I4\)
          CALL VCURXY(0,IR)
          CALL FILECH(FILIN,IER)
          IF(IER) GO TO 20
          IR = IR + 2
          CALL VCURXY(0,IR)
          WRITE(*,9002)
 9002  FORMAT(1X,'NO .SDF EXTENSION - PROGRAM ABORT!!')
         STOP
C
C HERE IF OK
C
 20      CONTINUE
         IR = IR + 2
         KEY = 'ENTER OUTPUT FILENAME - NO FILE EXTENSIONS PLEASE :$'
         CALL KEYBD(KEY,R,I,FILOT,3,(6),0,IR)
C
C VERIFY NO EXTENSION AND GET CURRENT LENGTH
C
         IFLEN = 6
         CALL FILEX(FILOT,IFLEN)
         NFLEN = IFLEN + 6
C
C NOW GET INPUT/OUTPUT CHANNELS
C
 30      CONTINUE
         IR = IR + 2
         CALL VCURXY(0,IR)
         KEY = 'ENTER THE NUMBER OF INPUT CHANNELS ( <= 10 ) :$'
         CALL KEYBD(KEY,R,NINPUT,A,2,(2),0,IR)
         IF(NINPUT.LE.10) GO TO 40
         IR = IR + 1
         CALL VCURXY(0,IR)
         WRITE(*,9030)
 9030  FORMAT(1X,'TOO MANY INPUT CHANNELS SPECIFIED'\)
         GO TO 30
 40      CONTINUE
         IR = IR + 2
         CALL VCURXY(0,IR)
         WRITE(*,9031)
 9031  FORMAT(1X,'SPECIFY MACRAN CHANNEL TO USE FOR EACH INPUT'\)
         IR = IR + 1
         CALL VCURXY(0,IR)
         KEY = '--> $'
         DO 45 IC=1,NINPUT
           IR = IR + 1
           CALL KEYBD(KEY,R,ICHNI(IC),A,2,(2),0,IR)
 45      CONTINUE
C
C NOW GET OUTPUT CHANNELS
C
 48      CONTINUE
         CALL VCLEAR
         IR = 2
         CALL VCURXY(0,IR)
         KEY = 'ENTER THE NUMBER OF OUTPUT CHANNELS ( <= 14 ) :$'
         CALL KEYBD(KEY,R,NOUTPT,A,2,(2),0,IR)
         IF(NOUTPT.LE.14) GO TO 50
         IR = IR + 1
         CALL VCURXY(0,IR)
         WRITE(*,9040)
 9040  FORMAT(1X,'TOO MANY OUTPUT CHANNELS SPECIFIED'\)
         GO TO 48
C
C NOW GET OUTPUT CHANNELS
C
 50      CONTINUE
         IR = IR + 2
         CALL VCURXY(0,IR)
         WRITE(*,9041)
 9041  FORMAT(1X,'SPECIFY MACRAN CHANNEL FOR EACH OUTPUT'\)
          IR = IR + 1
          CALL VCURXY(0,IR)
          KEY = '--> $'
          DO 55 IC = 1,NOUTPT
            IR = IR + 1
            CALL KEYBD(KEY,R,ICHNO(IC),A,2,(2),0,IR)
 55       CONTINUE
C
C SEE IF A DAMAGED OR UNDAMAGED CASE
C
          IR = IR + 2
          CALL VCURXY(0,IR)
 54       CONTINUE
          KEY = 'IS THIS A (D)AMAGED OR (U)NDAMAGED RUN? :$'
          CALL KEYBD(KEY,R,I,ATYPE,3,(1),0,IR)
          IF(ATYPE.EQ.'D' .OR. ATYPE.EQ.'d') GO TO 56
          IF(ATYPE.EQ.'U' .OR. ATYPE.EQ.'u') GO TO 56
          GO TO 54
 56       CONTINUE
C
C GET TITLE FOR THIS RUN
C
          KEY = 'ENTER A DESCRIPTIVE TITLE FOR THIS RUN : $'
          IR = IR + 2
          CALL KEYBD(KEY,R,I,TITLE,3,(30),0,IR)
C
C  NOW START CREATING THE BATCH FILE
C
          OPEN(6,FILE='NASAD.BAT',STATUS='NEW')
          OPEN(7,FILE='NASAREN.BAT',STATUS='NEW')
          OPEN(8,FILE='MATRX.DAT',STATUS='NEW')
          WRITE(8,4010) NINPUT,NOUTPT,ATYPE,TITLE
 4010  FORMAT(I5,2X,I5,2X,A1/A30)
C
C ASSIGN GENERIC FILENAMES TO MACRAN FILES
C
          MFIL1 = 'TRNSD1'
          MFIL2 = 'TRNSD2'
          MFIL3 = 'TRNSD3'
          MFIL4 = 'TRNSD4'
          FX1 = 'INP'
          FX2 = 'OUT'
          FX3 = 'SDF'
          FX4 = 'DAT'
          IL = 6
C
C NOW LOOP OVER THE NUMBER OF OUTPUT CHJANNELS TO PROCESS
C
          DO 100 ICHN=1,NOUTPT
C
C GET FIRST MACRAN FILENAME
C
            CALL FILEM(MFIL1,MFIL1I,FX1,ICHN,IL)
            OPEN(10,FILE=MFIL1I,STATUS='NEW')
C
C NOW WRITE OUT FIRST MACRAN CARD
C
            CALL MACRD1(FILIN,NINPUT,ICHNI,ICHNO(ICHN),TITLE,NPNTS)
            CLOSE(10,STATUS='KEEP')
            CALL FILEM(MFIL1,MFIL1O,FX2,ICHN,IL)
C
```

```
C GET SECOND MACRAN FILENAME
C
            CALL FILEM(MFIL2,MFIL2I,FX1,ICHN,IL)
            OPEN(10,FILE=MFIL2I,STATUS='NEW')
C
C WRITE OUT SECOND CARD
C
            CALL MACRD2
            CLOSE(10,STATUS='KEEP')
            CALL FILEM(MFIL2,MFIL2O,FX2,ICHN,IL)
C
C GET THIRD MACRAN FILENAME
C
            CALL FILEM(MFIL3,MFIL3I,FX1,ICHN,IL)
            OPEN(10,FILE=MFIL3I,STATUS='NEW')
C
C WRITE OUT THIRD MACRAN CARD
C
            CALL MACRD3
            CLOSE(10,STATUS='KEEP')
            CALL FILEM(MFIL3,MFIL3O,FX2,ICHN,IL)
C
C NOW ADD TO THE NASAD.BAT FILE
C
            WRITE(6,6010) MFIL1I,MFIL1O,MFIL1O
            WRITE(6,6060) MFIL2I,MFIL2O,MFIL2O
            WRITE(6,6010) MFIL3I,MFIL3O,MFIL3O
C
C GET OUTPUT FILENAME
C
            CALL FILEM(FILOT,FILOT1,FX3,ICHN,IFLEN)
            CALL FILEM(FILOT,FILOT2,FX4,ICHN,IFLEN)
C
C WRITE OUT BATCH COMMANDS
C
            WRITE(6,6020) FILOT1,FILOT1
            WRITE(6,6030) FILOT2,FILOT2
          WRITE(8,4020) (FILET(IFM),IFM=1,NFLEN)
 100      CONTINUE
 4020  FORMAT(12A1)
C
          IF(IFLEN.EQ.6) GO TO 150
          CALL FILAD(FILOT,IFLEN)
 150      CONTINUE
C
C FINISH OFF BATCH COMMANDS FOR NASA.BAT
C
          WRITE(6,6070)
C
          CLOSE(6,STATUS='KEEP')
C
C NOW ADD TO NASAREN.BAT
C
          WRITE(7,6050) FILOT,FILOT
          CLOSE(7,STATUS='KEEP')
C
C END IT ALL
C
          CLOSE(7,STATUS='KEEP')
          CALL VCURXY(0,23)
          STOP
C
```

```
C FORMAT SECTION
C
 6010  FORMAT('MAC423 /R 10000 <',A12,' >D:',A12/'DEL D:',A12)
 6020  FORMAT('DEL',1X,A12/'REN TRNSMD2.SDF',1X,A12)
 6030  FORMAT('DEL',1X,A12/'REN HD2.DAT',1X,A12)
 6040  FORMAT('DEL',1X,A6,'01.BFT'
     1  /'REN FORDI.DAT',1X,A6,'01.BFT')
 6045  FORMAT('DEL',1X,A6,'01.DFT'
     1  /'REN FORD01.DAT',1X,A6,'01.DFT')
 6046  FORMAT('DEL',1X,A6,'02.DFT'
     1  /'REN FORD02.DAT',1X,A6,'02.DFT')
 6050  FORMAT('DEL',1X,A6,'.BAT'
     1  /'REN NASAD.BAT',1X,A6,'.BAT')
 6060  FORMAT('MACTRN /R 10000 <',A12,' > D:',A12/'DEL D:',A12)
 6070  FORMAT('TESTRD.EXE'/'NASAREN.BAT')
          END
```

MACRD3.FOR

```
C
C SUBROUTINE MACCRD3 - PURPOSE TO WRITE OUT 3RD MACRAN CARD
C
         SUBROUTINE MACRD3
         IMPLICIT INTEGER*2 (I-N)
         WRITE(10,9010)
 9010  FORMAT('FILIN  TRNSMD2.SDF'
     1  /'FILOUT HD2.DAT'
     2  /'CTAPE',52X,'-1'
     3  /'END'
     4  /)
         RETURN
         END
```

MACRD2.FOR

```
C
C SUBROUTINE MACCRD2 - PURPOSE TO WRITE OUT 2ND MACRAN CARD
C
        SUBROUTINE MACRD2
        IMPLICIT INTEGER*2 (I-N)
C
C WRITE OUT STUFF
C
          WRITE(10,9010)
 9010  FORMAT('FILIN  MATRXD2.SDF'
     1       /'FILOUT TRNSMD2.SDF'
     2  /'TRANS  TAPE'
     3  /'END'
     4  /)
          RETURN
          END
```

MACRD1.FCR

```
C
C SUBROUTINE MACCRD1 - PURPOSE TO WRITE OUT FIRST MACRAN CARD
C
          SUBROUTINE MACRD1(FILIN,NINPUT,ICHNI,ICHNO,TITLE,NPNTS)
          IMPLICIT INTEGER*2 (I-N)
          DIMENSION ICHNI(10)
          CHARACTER*12 FILIN
          CHARACTER*30 TITLE
C
C WRITE OUT FIRST COUPLE OF COMMENT CARDS
C
          WRITE(10,9010)  TITLE
 9010  FORMAT('JBHEAD ',A30,5X,'TRANSFER FUNCTIONS')
          WRITE(10,9020) FILIN
 9020  FORMAT('FILIN',2X,A12)
C
C NOW WRITE OUT CHANNEL NUMBERS
C
          IF(NINPUT.GT.2) GO TO 50
          WRITE(10,9030) ICHNO,(ICHNI(IL),IL=1,NINPUT)
          GO TO 100
 50       CONTINUE
          WRITE(10,9030) ICHNO,(ICHNI(IL),IL=1,2)
          NLINES = (NINPUT+1)/4
          DO 60 IJ=1,NLINES
            IL = 3 + 4 * (IJ-1)
            IK = IL + 3
            IF(IK.GT.NINPUT) IK = NINPUT
            WRITE(10,9040) (ICHNI(IM),IM=IL,IK)
 60       CONTINUE
 9030  FORMAT('FILIN',1X,'CCHAN',14X,'1.',1X,3(9X,I2,'.'))
 9040  FORMAT('FILIN',1X,'CCHAN',5X,4(9X,I2,'.'))
C
C HERE FOR REST OF DATA
C
 100      CONTINUE
          WRITE(10,9050) NPNTS
 9050  FORMAT(
     1 'FILOUT MATRXD2.SDF'
     2/'PSD    TRANS',11X,I4,'.',
     3/'PSD    TOUTMTFC'
     4/'PSD    RETAIN'
     5/'END'
     6/)
          RETURN
          END
```

FILMAP.FOR

```
C
C SUBROUTINE FILMAP - TO CONSTRUCT A FILENAME
C                CALLED FORM FILEM
C
      SUBROUTINE FILMAP(FIL1,FIL2,FX,ANUM,ILEN)
      IMPLICIT INTEGER*2 (I-N)
      CHARACTER*1 ANUM(2),FX(3),FIL1(12),FIL2(12)
C
C
C
      DO 10 I=1,ILEN
       FIL2(I) = FIL1(I)
 10   CONTINUE
      DO 15 I=1,2
        FIL2(ILEN+I) = ANUM(I)
 15   CONTINUE
      FIL2(ILEN+3) = '.'
      DO 20 I=1,3
        FIL2(ILEN+3+I) = FX(I)
 20   CONTINUE
C
C CHANGE BLANKS TO ZEROES
C
      NLEN = ILEN + 6
      DO 30 I=1,NLEN
        IF(FIL2(I).EQ.' ') FIL2(I) = '0'
 30   CONTINUE
      RETURN
      END
```

FILEX.FOR

```
C
C SUBROUTINE TO CHECK ON FILE AND ITS LENGTH
C
         SUBROUTINE FILEX(FIL1,ILEN)
         IMPLICIT INTEGER*2 (I-N)
         CHARACTER*1 FIL1(20)
C
C GET LENGTH OF FILE
C
         DO 5 I=1,ILEN
           IF(FIL1(I).EQ.' ') GO TO 7
 5       CONTINUE
         GO TO 8
 7       CONTINUE
         ILEN = I-1
C
C NOW SEE IF AN EXTENSION WAS ADDED
C
 8       CONTINUE
         DO 10 I=1,ILEN
           IF(FIL1(I).EQ.'.') GO TO 20
 10      CONTINUE
         RETURN
 20      CONTINUE
         ILEN = ILEN-1
         RETURN
         END
```

FILEM.FOR

```
C
C SUBROUTINE FILEM - TO CONSTRUCT A FILENAME
C
          SUBROUTINE FILEM(FIL1,FIL2,FX,INUM,ILEN)
          IMPLICIT INTEGER*2 (I-N)
          CHARACTER*1 FIL1(12),FIL2(12),FX(3)
          CHARACTER*2 ANUM
C
C WRITE OUT INUM TO ANUM
C
          WRITE(ANUM,900) INUM
 900  FORMAT(I2)
C
C CALL FILCHG TO GET FILENAME
C
          CALL FILMAP(FIL1,FIL2,FX,ANUM,ILEN)
          RETURN
          END
```

FILECH.FOR

```
C
C SUBROUTINE FILECH - PURPOSE TO SEE IF A .SDF EXTENSION IS ON
C
          SUBROUTINE FILECH(FIL1,IER)
          IMPLICIT INTEGER*2 (I-N)
          CHARACTER*1 FIL1(12)
          LOGICAL*2 IER
C
C FIRST LOOK FOR A .
C
          IER = .FALSE.
          DO 10 I=1,12
            IF(FIL1(I).EQ.'.') GO TO 20
 10       CONTINUE
          RETURN
 20       CONTINUE
          ILEN = I
          IF(FIL1(ILEN+1).NE.'S' .AND. FIL1(ILEN+1).NE.'s')RETURN
          IF(FIL1(ILEN+2).NE.'D' .AND. FIL1(ILEN+2).NE.'d')RETURN
          IF(FIL1(ILEN+3).NE.'F' .AND. FIL1(ILEN+3).NE.'f')RETURN
          IER = .TRUE.
          RETURN
          END
```

FILBD.FOR

```
$NOFLOATCALLS
          CHARACTER*80 KEY
          CHARACTER*14 FLNAME
C
C
C
          CALL VCLEAR
          CALL VCURXY(0,0)
          KEY = 'ENTER FILENAME TO DISPLAY : $'
          CALL KEYBD(KEY,R,I,FLNAME,3,(14),0,5)
          OPEN(7,FILE=FLNAME,FORM='UNFORMATTED')
 10       CONTINUE
          READ(7) FTR,FTI
          WRITE(*,8000) FTR,FTI
 8000  FORMAT(2X,1PE10.3,5X,1PE10.3)
          GO TO 10
          STOP
          END
```

FILAD.FOR

```
C
C SUBROUTINE FILAD(FILOT,IFLEN)
C
        SUBROUTINE FILAD(FILOT,IFLEN)
        IMPLICIT INTEGER*2 (I-N)
        CHARACTER*1 FILOT(6)
        IFL = IFLEN + 1
        DO 10 I=IFL,6
          FILOT(I) = '0'
 10     CONTINUE
        RETURN
        END
```

```
A:\MTFM\DAMAGE
>DIR

 Volume in drive A has no label
 Directory of  A:\MTFM\DAMAGE


MACWRT   FOR      1575   1-07-87    9:58a
DAMCOM   FOR      1096   1-27-87    2:04p
DAMAGE   FOR      2857   1-27-87    2:07p
```

MACWRT.FOR

```
$NOFLOATCALLS
          SUBROUTINE MACWRT(DAM,NFREQ,DELF,TITLE)
          IMPLICIT INTEGER*2 (I-N)
          COMPLEX DAM(7,2500),XDAM(7)
          REAL*4 RDAM(16)
          CHARACTER*4 ADAM(7)
          CHARACTER*30 TITLE
C
C SET UP CHARACTER STRINGS
C
          ADAM(1) = 'DAM1'
          ADAM(2) = 'DAM2'
          ADAM(3) = 'DAM3'
          ADAM(4) = 'DAM4'
          ADAM(5) = 'DAM5'
          ADAM(6) = 'DAM6'
          ADAM(7) = 'DAM7'
C
C OPEN UP OUTPUT DAMAGE.SCF FILE
C
          OPEN(20,FILE='DAMAGE.SCF',STATUS='NEW')
C
C START WRITING OUT TO FILE
C
          WRITE(20,9090) TITLE
 9090  FORMAT('JBHEAD',1X,'COMPUTED DAMAGE - ',A30)
          WRITE(20,9050)
          WRITE(20,9000)
          WRITE(20,9010)
          WRITE(20,9020)
          DO 20 I=1,6
            DO 10 IW=1,2
              WRITE(20,9030) ADAM(I)
 10         CONTINUE
 20       CONTINUE
          DO 30 I=1,16
            RDAM(I) = 0.0
 30       CONTINUE
          CALL GCURXY(IC,IR)
          IR = IR + 1
          CALL VCURXY(0,IR)
          DO 100 IFQ=1,NFREQ
            RDAM(1) = FLOAT(IFQ-1) * DELF
              DO 40 I=1,6
                XDAM(I) = DAM(I,IFQ)
                RI = AIMAG(XDAM(I))
                RR = REAL(XDAM(I))
                RDAM(I*2) = RR
                RDAM(I*2+1) = RI
 40           CONTINUE
              WRITE(20,9040) (RDAM(I),I=1,16)
 100      CONTINUE
          WRITE(20,9060)
          CLOSE(20,STATUS='KEEP')
 9000  FORMAT('CTAPE',2X,'NASA 1311.05')
 9010  FORMAT('CTAPE',2X,'DAMAGE ',16X,'13')
 9020  FORMAT('CTAPE',2X,'FREQUENCY')
 9030  FORMAT('CTAPE',2X,A4)
 9040  FORMAT('CTAPE',19X,4(E11.3,1X),
     1          /'CTAPE C',17X,4(E11.3,1X),
     2          /'CTAPE C',17X,4(E11.3,1X),
     3          /'CTAPE C',17X,4(E11.3,1X))
 9050  FORMAT('FILOUT DAMAGE.SDF')
 9060  FORMAT('END')
          RETURN
          END
```

DAMCOM.FOR

```
$NOFLOATCALLS
      SUBROUTINE DAMCOM(IFQ)
      IMPLICIT INTEGER*2 (I-N)
      COMPLEX*8 TFU(14,10),TFD(14,10)
      COMPLEX*8 DAM(7,2500),AUX1,AUX2,AUX3,AUX4,AUX5
      DIMENSION ILMI(14),ILMO(14)
      COMMON/DAMC/TFU,TFD,DAM,ILMI,ILMO,NINPUT,NOUTPT
C
C ZERO OUT AUX
C
      AUX1 = (0.,0.)
      AUX2 = (0.,0.)
      AUX3 = (0.,0.)
      AUX4 = (0.,0.)
      AUX5 = (0.,0.)
C
C COMPUTE DAMAGE
C
      DO 20 KK=1,NOUTPT
        DO 19 IK=1,NOUTPT
          IF(KK.EQ.ILMO(IK)) GO TO 20
 19     CONTINUE
        DO 10 JJ=1,NINPUT
          DO 11 IJ=1,NINPUT
            IF(JJ.EQ.ILMI(IJ)) GO TO 10
 11       CONTINUE
C         AUX1 = AUX1 + TFU(KK,JJ) * FTO(KK) * FTI(JJ)
C         AUX2 = AUX2 + TFD(KK,JJ) * FTO(KK) * FTI(JJ)
          AUX3 = AUX3 + TFU(KK,JJ) * TFU(KK,JJ)
          AUX4 = AUX4 + TFU(KK,JJ) * TFD(KK,JJ)
          AUX5 = AUX5 + TFD(KK,JJ) * TFD(KK,JJ)
 10     CONTINUE
 20   CONTINUE
      DAM(1,IFQ) = (AUX5 + AUX4) / (AUX4 + AUX3)
      DAM(2,IFQ) = (AUX4 + AUX3) / (AUX5 + AUX4)
      DAM(3,IFQ) = AUX3/AUX4
      DAM(4,IFQ) = AUX4 / AUX3
      DAM(5,IFQ) = AUX4 / AUX5
      DAM(6,IFQ) = AUX5 / AUX4
      RETURN
      END
```

DAMAGE.FOR

```
$NOFLOATCALLS
      IMPLICIT INTEGER*2 (I-N)
      COMPLEX*8 TFU(14,10),TFD(14,10)
      COMPLEX*8 DAM(7,2500)
      DIMENSION ILMI(14),ILMO(14)
      CHARACTER*80 KEY
      CHARACTER*30 TITLE
      CHARACTER*20 TFUF,TFDF
      CHARACTER*1 ATYPE
      COMMON/DAMC/TFU,TFD,DAM,ILMI,ILMO,NINPUT,NOUTPT
C
C
C
      CALL VCLEAR
      CALL VCURXY(0,0)
      IR = 0
C
C GET EACH FILENAME,OPEN IT UP AND READ HEADER INFO
C
      TFUF = 'HWUMAT.DAT'
      OPEN(6,FILE=TFUF,FORM='UNFORMATTED')
      READ(6) TITLE,NFREQ,NINPUT,NOUTPT,ATYPE,DELF
      WRITE(*,9000) TITLE,NINPUT,NOUTPT,NFREQ,DELF
 9000 FORMAT(1X,A30
     1 /1X,'NUMBER OF INPUT CHANNELS          : ',I4
     2 /1X,'NUMBER OF OUTPUT CHANNELS         : ',I4
     3 /1X,'NUMBER OF FREQUENCY POINTS        : ',I4
     4 /1X,'FREQUENCY STEP SIZE               : ',1PE10.3,' HZ'\)
      IR = IR + 6
      CALL VCURXY(0,IR)
      IF(ATYPE.EQ.'U' .OR. ATYPE.EQ.'u') WRITE(*,9010)
      IF(ATYPE.EQ.'D' .OR. ATYPE.EQ.'d') WRITE(*,9015)
 9010 FORMAT(1X,'UNDAMAGED'\)
 9015 FORMAT(1X,'DAMAGED'\)
      IR = IR + 2
      CALL VCURXY(0,IR)
C
C GET DAMAGED TF FILENAME
C
      TFDF = 'HWDMAT.DAT'
      OPEN(7,FILE=TFDF,FORM='UNFORMATTED')
      READ(7) TITLE,NFREQ,NINPUT,NOUTPT,ATYPE,DELF
      WRITE(*,9000) TITLE,NINPUT,NOUTPT,NFREQ,DELF
      IR = IR + 6
      CALL VCURXY(0,IR)
      IF(ATYPE.EQ.'U' .OR. ATYPE.EQ.'u') WRITE(*,9010)
      IF(ATYPE.EQ.'D' .OR. ATYPE.EQ.'d') WRITE(*,9015)
      IR = IR + 2
C
C SEE ABOUT ROW/COLUMN ELIMINATION
C
      PAUSE
      CALL VCLEAR
      IR = 0
      CALL VCURXY(0,IR)
      DO 40 I=1,14
        ILMI(I) = 0
        ILMO(I) = 0
 40   CONTINUE
```

```
          KEY = 'HOW MANY INPUT CHANNELS DO YOU WANT TO ELIMINATE? : $'
          CALL KEYBD(KEY,R,IELMI,A,2,(2),0,IR)
          IR = IR + 1
          IF(IELMI.EQ.0) GO TO 75
          CALL VCURXY(0,IR)
          WRITE(*,9200)
9200   FORMAT(1X,'ENTER CHANNELS TO ELIMINATE'\)
          IR = IR + 1
          CALL VCURXY(0,IR)
          DO 50 I=1,IELMI
            KEY = '--> $'
            CALL KEYBD(KEY,R,ILMI(I),A,2,(2),0,IR)
            IR = IR + 1
50        CONTINUE
75        CONTINUE
          IR = IR + 1
          KEY='HOW MANY OUTPUT CAHNNELS DO YOU WANT TO ELIMINATE? : $'
          CALL KEYBD(KEY,R,IELMO,A,2,(2),0,IR)
          IF(IELMO.EQ.0) GO TO 100
          IR = IR + 1
          CALL VCURXY(0,IR)
          WRITE(*,9200)
          IR = IR + 1
          CALL VCURXY(0,IR)
          DO 80 I=1,IELMO
            KEY = '--> $'
            CALL KEYBD(KEY,R,ILMO(I),A,2,(2),0,IR)
            IR = IR + 1
80        CONTINUE
100       CONTINUE
C
C NOW READ IN ALL THE DATA, ONE FREQ POINT AT A TIME AND COMPUTE
C DAMAGE FOR THAT FREQ POINT
C
          NTOT = NINPUT + NOUTPT
          DO 150 IFQ=1,NFREQ
            DO 110 IO=1,NOUTPT
            READ(6) (TFU(IO,I),I=1,NINPUT)
            READ(7) (TFD(IO,I),I=1,NINPUT)
110       CONTINUE
          CALL DAMCOM(IFQ)
150       CONTINUE
C
C CALL MACWRT TO WRITE OUT DATA
C
          CALL MACWRT(DAM,NFREQ,DELF,TITLE)
C
C END IT ALL
C
          CALL GCURXY(IC,IR)
          IR = IR + 1
          CALL VCURXY(0,IR)
          STOP
          END
```

```
A:\MTFM\TESTRD
>DIR

 Volume in drive A has no label
 Directory of  A:\MTFM\TESTRD


TESTRD   FOR      1591    1-27-87   11:[illegible]
```

TESTRD.FOR

```
$NOFLOATCALLS
          IMPLICIT INTEGER*2 (I-N)
          COMPLEX*8 HW(30),FT(30)
          DIMENSION ADAT(30),ODAT1(30),ODAT2(30),FREQD(2)
          CHARACTER*12 FLNAME
          CHARACTER*1 ADUM,ATYPE
          CHARACTER*30 TITLE
C
C READ IN DATA FROM FILE
C
          OPEN(8,FILE='MATRX.DAT')
          READ(8,5000) NINPUT,NOUTPT,ATYPE,TITLE
 5000   FORMAT(I5,2X,I5,2X,A1/A30)
C
C NOW READ IN EACH FILENAME AND OPEN UP A CHANNEL
C
          DO 2 I=1,NOUTPT
            READ(8,5010) FLNAME
            ICHN = I + 20
            OPEN(ICHN,FILE=FLNAME)
 2        CONTINUE
 5010   FORMAT(A12)
          CLOSE(8,STATUS='KEEP')
C
C OPEN UP OUTPUT FILE
C
          IF(ATYPE.EQ.'U' .OR. ATYPE.EQ.'u') GO TO 4
          OPEN(6,FILE='HWDMAT.DAT',STATUS='NEW',FORM='UNFORMATTED')
          GO TO 5
 4        CONTINUE
          OPEN(6,FILE='HWUMAT.DAT',STATUS='NEW',FORM='UNFORMATTED')
 5        CONTINUE
C
C READ IN FIRST RECORD AND GET THE NUMBER OF FREQ RECORDS
C
          READ(21,9000) NREC
 9000   FORMAT(/30X,I5)
          NFREC = (NREC-1)/2
          NRECD = NREC + 2
          REWIND 21
          NLINE = (NREC+3)/4
C
C READ CHANNEL 21 AND COMPUTE # OF FREQ POINTS
C
          DO 11 I=1,NRECD
            READ(21,9001) ADUM
  11      CONTINUE
          NFREQ = 0
 12       CONTINUE
          DO 13 I=1,NLINE
            READ(21,9001,END=14) ADUM
 13       CONTINUE
          NFREQ = NFREQ + 1
          GO TO 12
 14       CONTINUE
          REWIND 21
          WRITE(*,4010) NFREQ
 4010   FORMAT(1X,I5,' FREQUENCY POINTS READ')
```

```
C
C READ IN FIRST TWO FREQ VALUES AND COMPUTED DELTA FREQ
C
          DO 21 I=1,NRECD
            READ(21,9001) ADUM
 21       CONTINUE
          DO 22 I=1,2
            READ(21,5800) FREQD(I)
            DO 22 IL=2,NLINE
              READ(21,9001) ADUM
 22       CONTINUE
 5800  FORMAT(24X,E11.3)
          REWIND 21
          DELF = FREQD(2) - FREQD(1)
          WRITE(*,5810) DELF
 5810  FORMAT(1X,'FREQUENCY STEP SIZE : ',1PE10.3,' HZ')
C
C READ OVER DUMMY DATA IN EACH FILE
C
          DO 6 IFL=1,NOUTPT
            ICHN = 20 + IFL
          DO 10 I=1,NRECD
            READ(ICHN,9001) ADUM
 10       CONTINUE
 6        CONTINUE
 9001  FORMAT(A1)
C
C NOW WRITE OUT HEADER INFO
C
          WRITE(6) TITLE,NFREQ,NINPUT,NOUTPT,ATYPE,DELF
C
C NOW LOOP OVER EACH FILE,READING A ROW AT EACH FREQ POINT
C
 20       CONTINUE
          DO 30 IFL=1,NOUTPT
            ICHN = IFL + 20
          DO 15 IW=1,NLINE
            II = (IW-1) * 4 + 1
            JJ = II + 3
            READ(ICHN,9010,END=100) (ADAT(K),K=II,JJ)
 15       CONTINUE
C
C NOW PLACE IN COMPLEX ARRARY
C
          DO 18 IW=1,NFREC
            HW(IW) = CMPLX(ADAT(IW*2),ADAT(IW*2+1))
 18       CONTINUE
          WRITE(6) (HW(I),I=1,NFREC)
 30       CONTINUE
 9010  FORMAT(24X,4(E11.3,1X))
          GO TO 20
 100      CONTINUE
          DO 105 IFL=1,NOUTPT
            ICHN = 20 + IFL
            CLOSE(ICHN,STATUS='KEEP')
 105      CONTINUE
          STOP
          END
```

```
A:\MTFM\INTRFS
>DIR

 Volume in drive A has no label
 Directory of  A:\MTFM\INTRFS


OUT1      [illegible]      1478   1-20-87    8:[illegible]a
INTRFS    [illegible]      7660   1-07-87   10:[illegible]a
```

OUT1.FOR

```
$NOFLOATCALLS
          SUBROUTINE OUT1
          PARAMETER(NTIME=5000)
          IMPLICIT INTEGER*2 (I-N)
          DIMENSION XIN(24,NTIME)
          CHARACTER*12 MACOUT,MACBIN,MACIN
          CHARACTER*8 CHANDES(24)
          COMMON/NASA/XIN,CHANDES,NTOT,NPTS,DT,MACIN,MACOUT,MACBIN
          NREC = NTOT + 1
C
C WRITE OUT FIRST SET OF CTAPE CARDS
C
          OPEN(4,FILE=MACOUT,STATUS='NEW')
          WRITE(4,9001) MACIN
 9001  FORMAT('JBHEAD',1X,'TIME HISTORIES FROM COSMOS FILE - ',A12)
          WRITE(4,9000) MACBIN
 9000  FORMAT('FILOUT ',A12)
          WRITE(4,9010)
 9010  FORMAT('CTAPE',2X,'NASA  1311.05')
          WRITE(4,9020) NREC
 9020  FORMAT('CTAPE',2X,'COSMOS/M',15X,I2,'.')
          WRITE(4,9030)
 9030  FORMAT('CTAPE',2X,'TIME')
          DO 10 I=1,NTOT
            WRITE(4,9040) CHANDES(I)
 10       CONTINUE
 9040  FORMAT('CTAPE',2X,A8)
C
C WRITE OUT DATA
C
          IF(NREC.GT.4) GO TO 200
C
C HERE IF LESS THAN 4 OUTPUT CHANNELS
C
          DO 110 I=1,NPTS
            TIME = (I-1) * DT
            WRITE(4,9050) TIME,(XIN(IFN,I),IFN=1,NTOT)
 110      CONTINUE
          WRITE(4,9090)
          RETURN
 200      CONTINUE
          NREC = NREC-4
          NLINE = (NREC+3)/4
          DO 210 I=1,NPTS
            TIME = (I-1) * DT
            WRITE(4,9050) TIME,(XIN(IFN,I),IFN=1,3)
                DO 215 IP = 1,NLINE
                  II = (IP-1) * 4 + 4
                  JJ = II + 3
                  IF(JJ.GT.NTOT) JJ = NTOT
                  WRITE(4,9060) (XIN(IFN,I),IFN=II,JJ)
 215            CONTINUE
 210      CONTINUE
          WRITE(4,9090)
 9090  FORMAT('END')
 9050  FORMAT('CTAPE',19X,4(1PE11.3,1X))
 9060  FORMAT('CTAPE C',17X,4(1PE11.3,1X))
          RETURN
          END
```

INTRFS.FOR

```
C*********************************************************************
C     THIS PROGRAM TAKES THE OUTPUT DATA (*.PLT) FILE FROM COSMOS    *
C            AND PREPARES AN INPUT FILE (*.SCF) FOR MACRAN.          *
C                                                                    *
C                    NASA PROJECT JOB # 1311.05B                     *
C                                                                    *
C     -----------------------------------------------------------    *
C                                                                    *
C     THE PROGRAM WILL OPEN A MAXIMUM OF 25 CHANNELS. TIME WILL      *
C     ALWAYS BE CHANNEL 1 FOLLOWED BY 10 INPUT CHANNELS AND THE      *
C     OUTPUT CHANNELS( MAXIMUM = 14 ).  THE PROGRAM ALSO CREATES A   *
C     BATCH FILE NAMED NASAMAC.BAT, WHICH UPON EXECUTION, BESIDES    *
C     PERFORMING THE AFOREMENTIONED TASK WILL CALL MACRAN AND        *
C     CONVERT THE ASCII(*.SCF) FILE TO A BINARY(*.SDF) FILE.         *

C                                                                    *
C     ------------------------------------------------------------   *
C                                                                    *
C                    DEFINITION OF VARIABLES                         *
C                                                                    *
C                    TIME     : DISCRETE TIME SERIES                 *
C                    XLOD     : INPUTS FROM COSMOS                   *
C                    XOUT     : OUTPUTS FROM COSMOS                  *

C                    NTIME    : PARAMETER FOR VARIABLE DIMENSION,    *
C                             : SET TO MAXIMUM OF 5000 POINTS.       *
C                    NPTS     : NO OF POINTS IN THE TIME HISTORY     *
C                    ICHANO   : NO. OF OUTPUT CHANNELS               *
C                    ICHANI   : NO. OF INPUT CHANNELS                *
C                                                                    *
C     ------------------------------------------------------------   *
C                                                                    *
C                              USER INPUTS                           *
C                                                                    *
C                    NAME OF THE INPUT FILE NAME      : (*.PLT)      *
C                    NAME OF THE MACRAN ASII FILE     : (*.SCF)      *
C                    NAME OF THE MACRAN BINARY FILE   : (*.SDF)      *
C                    NO OF POINTS IN THE TIME HISTORY : NPTS         *
C                                                                    *
C     ------------------------------------------------------------   *
C                                                                    *
C                                 NOTES                              *
C                                                                    *
C     1. THE SOURCE CODE HAS BEEN PROGRAMMED TO ACCEPT 5000 TIME     *
C        POINTS, MADE AVAILABLE TO IT FROM COSMOS PLOT FILE          *
C        (*.PLT).  SHOULD IT BE NECESSARY TO INCREASE THIS LIMIT     *
C        CHANGE THE VALUE OF 'NTIME=' APPEARING IN THE PARAMETER     *
C        STATETMENT OF THE MAIN PROGRAM -- REFORM, AND THE FIVE      *
C        SUBROUTINES IT CALLS NAMELY; OUT1, OUT2, OUT3, OUT4,        *
C        AND OUT5.  THIS HAS TO BE DONE BY EDITING THE SOURCE        *
C        PROGRAM --- INTRFS.FOR .                                    *
C                                                                    *
C     2. ITS IMPERATIVE THAT THE USER ACQUAINTS HIMSELF WITH THE     *
C        COSMOS PLOT FILE AND IS FAMILIAR WITH THE INPUTS AND THE    *
C        OUTPUTS SPECIFIED IN THE COSMOS RUN.  THE CODE IS WRITTEN   *
C        FOR TEN INPUTS(STRESSES, DISPLACEMENTS).  IT CONSIDERS      *
C        ACCELERATION AS ITS OUTPUTS ONLY.  IT IS RECOMMENDED        *
C        THAT THE USER MAKES A NOTE OF THESE PARAMETERS AND          *
C        UTILIZES ONLY THE NON-ZERO QUANTITIES IN THE SECOND MACRO.  *
C        THIS WILL PRECLUDE ERRONEOUS TRANSFER FUNCTIONS RESULTING   *
C        FROM DIVISION OR MULTIPLICATION OF ZERO.                    *
C                                                                    *
C     3. TO EXECUTE THIS PROGRAM RUN --- NASAMAC.BAT                 *
C                                                                    *
C*********************************************************************
```

```
$NOFLOATCALLS
      PARAMETER(NTIME=5000)
          IMPLICIT INTEGER*2 (I-N)
      CHARACTER*120 ALINE
      CHARACTER*12 MACIN,MACOUT,MACBIN
          CHARACTER*4 ATIME
          CHARACTER*3 AYN
          CHARACTER*80 KEY
          DIMENSION XIN(24,NTIME)
          CHARACTER*8 CHANDES(24)
          COMMON/NASA/XIN,CHANDES,NTOT,NPTS,DT,MACIN,MACOUT,MACBIN
C
C CLEAR SCREEN, DISPLAY PROMPT ,ETC
C
          CALL VCLEAR
          CALL VCURXY(0,0)
          KEY='INTRFS - PROGRAM TO REFORM COSMOS FILE TO MACRAN FILE $'
          CALL PRMPT(KEY,1.00)
          CALL GCURXY(IC,IR)
          IR = IR + 1
C
C GET INPUT AND OUTPUT FILENAMES
C
          KEY = 'ENTER COSMOS PLOT FILENAME TO PROCESS : $'
          CALL KEYBD(KEY,R,I,MACIN,3,(12),0,IR)
          IR = IR + 2
          KEY = 'ENTER OUTPUT MACRAN .SCF FILENAME - INCLUDE .SCF : $'
          CALL KEYBD(KEY,R,I,MACOUT,3,(12),0,IR)
          IR = IR + 2
          KEY = 'ENTER OUTPUT MACRAN .SDF FILENAME - INCLUDE .SDF : $'
          CALL KEYBD(KEY,R,I,MACBIN,3,(12),0,IR)
C
C GET THE NUMBER OF POINT IN THE TIME HISTORY
C
          IR = IR + 2
          KEY = 'ENTER THE NUMBER OF POINTS IN THE TIME HISTORY : $'
          CALL KEYBD(KEY,R,NPTS,A,2,(4),0,IR)
          IR = IR + 1
          KEY = 'DO YOU WANT TO TRUNCATE THE NUMBER OF POINTS? $'
          CALL KEYBD(KEY,R,IYN,AYN,4,(3),0,IR)
          NEWPT = NPTS
          IF(IYN.EQ.0) GO TO 30
          IR = IR + 1
          KEY ='ENTER NEW NUMBER OF POINTS : $'
          CALL KEYBD(KEY,R,NEWPT,A,2,(4),0,IR)
  30      CONTINUE
C
C  OPEN UP INPUT COSMOS FILE AND .SCF MACRAN FILE
C
          OPEN(2,FILE=MACIN)
C
C GET INPUT AND OUTPUT CHANNEL ASSIGNMENTS
C
          IR = IR + 2
          KEY = 'ENTER THE NUMBER OF LOADS TO USE FROM COSMOS : $'
          CALL KEYBD(KEY,R,NFORC,A,2,(2),0,IR)
          IR = IR + 2
          IF(NFORC.NE.0) GO TO 20
          KEY = 'IS THERE LOAD DATA IN THE COSMOS PLOTFILE? : $'
          CALL KEYBD(KEY,R,IREWD,AYN,4,(3),0,IR)
          IR = IR + 2
  20      CONTINUE
```

```
C
C GET CHANNEL DESCRIPTIONS FOR EACH LOAD INPUT
C
          IF(NFORC.EQ.0) GO TO 8
          CALL VCURXY(0,IR)
            WRITE(*,8020)
 8020  FORMAT(1X,'ENTER A CHANNEL DESCRIPTION FOR EACH FORCE USED'\)
          CALL VCURXY(0,IR)
          DO 5 IW=1,NFORC
            IR = IR + 1
            CALL VCURXY(0,IR)
            KEY = '--> $'
            CALL KEYBD(KEY,R,I,CHANDES(IW),3,(8),0,IR)
 5        CONTINUE
 8        CONTINUE
          CALL VCLEAR
          CALL VCURXY(0,0)
          IR = 2
          KEY='ENTER THE NUMBER OF ACCELERATIONS TO USE FROM COSMOS : $'
          CALL KEYBD(KEY,R,NACC,A,2,(2),0,IR)
          IR = IR + 2
          CALL VCURXY(0,IR)
          WRITE(*,8030)
 8030 FORMAT(1X,'ENTER CHANNEL DESCRIPTION FOR EACH ACCELERATION USED'\)
          IP = NFORC + 1
          NTOT = NFORC + NACC
          DO 15 IW=IP,NTOT
            IR = IR + 1
             CALL VCURXY(0,IR)
            KEY = '--> $'
            CALL KEYBD(KEY,R,I,CHANDES(IW),3,(8),0,IR)
 15       CONTINUE
C
C READ PAST FIRST HEADER IN MACIN, COMPUTE DELTA T AND COMPUTE
C THE NUMBER OF POINTS IN THE FILE
C
          IR = IR + 2
          CALL VCURXY(0,IR)
          WRITE(*,7010) MACIN
 7010  FORMAT(1X,'READING COSMOS PLOTFILE : ',A12\)
          CALL VCURXY(0,IR)
          READ(2,26) ALINE
 26       FORMAT(A)
          READ(2,9010) TIME1
          READ(2,9010) TIME2
 9010 FORMAT(1X,E10.3)
          DT = TIME2 - TIME1
          WRITE(*,7000)
 7000  FORMAT(80X\)
          CALL VCURXY(0,IR)
          WRITE(*,7020) DT
 7020  FORMAT(1X,' FILE HAS A TIME STEP OF : ',1PE10.3,' SECONDS'\)
          IR = IR + 2
          CALL VCURXY(0,IR)
C
C NOW REWIND FILE AND READ IN DATA
```

```
C
          REWIND 2
          IF((NFORC.EQ.0).AND.(IREWD.EQ.0)) GO TO 100
          IF(NFORC.EQ.0) GO TO 50
          READ(2,26) ALINE
          WRITE(*,7040)
 7040  FORMAT(1X,'READING IN LOAD DATA FROM COSMOS PLOTFILE '\)
          IR = IR + 2
          CALL VCURXY(0,IR)
          DO 40 I=1,NPTS
            READ(2,9030) (XIN(IFN,I),IFN=1,NFORC)
 40    CONTINUE
          GO TO 100
 9030  FORMAT(12X,10(E10.3,1X))
 9040  FORMAT(12X,3(22X,E10.3,1X))
C
C HERE IF NO FORCES NEEDED BUT MUST BE READ OVER
C
 50       CONTINUE
          READ(2,26) ALINE
          WRITE(*,7050)
 7050  FORMAT(1X,'SKIPPING OVER LOAD DATA IN COSMOS PLOTFILE'\)
          IR = IR + 2
          CALL VCURXY(0,IR)
          DO 60 I=1,NPTS
            READ(2,26) ALINE
 60       CONTINUE
C
C HERE TO READ ACCELERATIONS, IF ANY
C
 100      CONTINUE
          IF(NACC.EQ.0) GO TO 200
          READ(2,26) ALINE
          WRITE(*,7060)
 7060  FORMAT(1X,'READING IN ACCELERATION DATA'\)
          IR = IR + 2
          CALL VCURXY(0,IR)
          NLINO = (NACC+2)/3
          IP = 1
 103      CONTINUE
            IAC1 = (IP-1) * 3 + 1 + NFORC
            IAC2 = IAC1 + 2
            IF(IAC2.GT.NTOT) IAC2 = NTOT
          DO 105 I=1,NPTS
            READ(2,9040) (XIN(IFN,I),IFN=IAC1,IAC2)
 105   CONTINUE
          IF(IP.EQ.NLINO) GO TO 110
          IP = IP + 1
          READ(2,26) ALINE
          GO TO 103
 110      CONTINUE
C
C HERE WHEN ALL DONE READING IN DATA
C TIME TO REFORMAT TO MACRAN -.SCF FORMAT
C
 200      CONTINUE
          WRITE(*,7070)
 7070  FORMAT(1X,'REFORMATTING TO COSMOS -.SCF FORMAT'\)
          IR = IR + 2
          CALL VCURXY(0,IR)
          NPTS = NEWPT
          CALL OUT1
C
```

```
C HERE TO WRITE OUT REST OF MACRO
C
      OPEN(6,FILE='NASAMAC.BAT',STATUS='NEW')
      WRITE(6,19)
19    FORMAT('INTRFS.EXE')
      WRITE(6,1911) MACOUT
1911  FORMAT('MAC423 /R 10000 <',A12,1X,'>MACOUT.OUT')
      WRITE(6,1912)
1912  FORMAT('DEL MACOUT.OUT')
      WRITE(6,1913)
1913  FORMAT('NASADAM')
      CLOSE(6)
C
C HERE WHEN DONE
C
          STOP
          END
```

MODAL STRAIN ENERGY DISTRIBUTION METHOD (MSEDM)
SOURCE CODE LISTINGS

```
A:\MSEDM
>DIR

 Volume in drive A has no label
 Directory of  A:\MSEDM


DAM10      FOR       1913    2-19-87   10:10a
MODSSUM    FOR        922    2-17-87    4:29p
MODSTRN    FOR       1647    2-06-87    [illegible]:49a
MODSUM     FOR       4242    2-23-87    [illegible]:29a
```

DAMIO.FOR

```
$NOFLOATCALLS
          IMPLICIT INTEGER*2 (I-N)
          DIMENSION DAM(20),DAMCRT(20)
          CHARACTER*80 KEY
          CHARACTER*60 DAMTTL
          CHARACTER*20 DAMFIL
          CHARACTER*1 IEDCR
C
C CLEAR SCREEN, ETC
C
          CALL VCLEAR
          CALL VCURXY(0,0)
          KEY = 'DAMIO - PROGRAM TO CREATE/EDIT DAMAGE PARAMETERS$'
          CALL PRMPT(KEY,1.00)
          IR = 6
          IR = IR + 1
          KEY = 'DO YOU WANT TO (C)REATE OR (E)DIT A DAMAGE FILE? : $'
          CALL KEYBD(KEY,R,I,IEDCR,3,(1),0,IR)
C
C HERE FOR FILE CREATION
C
          CALL VCLEAR
          IR = 0
          CALL VCURXY(0,IR)
          KEY = 'DAMAGE TITLE : $'
          CALL KEYBD(KEY,R,I,DAMTTL,3,(60),0,IR)
          IR = IR + 1
          KEY = 'NUMBER OF MODES USED : $'
          CALL KEYBD(KEY,R,NMODES,A,2,(2),0,IR)
          IR = IR + 2
          CALL VCURXY(0,IR)
          WRITE(*,9010)
 9010  FORMAT(1X,'MODE',5X,'D(W)',10X,'Dcrit(W)'\)
          IR = IR + 1
          CALL VCURXY(0,IR)
          WRITE(*,9002)
 9002  FORMAT(' NO.'\)
          IR = IR + 1
C
C NOW READ IN DATA
C
          IST = IR
          IEND = IR + 9
          DO 20 IM=1,NMODES
          CALL VCURXY(0,IR)
            IF(IR.EQ.IST) WRITE(*,9200) IM
            IF(IR.NE.IST) WRITE(*,9201) IM
            KEY = ' $'
            CALL KEYBD(KEY,DAM(IM),I,A,1,(10),9,IR)
            CALL KEYBD(KEY,DAMCRT(IM),I,A,1,(10),23,IR)
            IR = IR + 1
            IF(IR.LE.IEND) GO TO 20
            DO 15 IER=IST,IEND
                CALL VCURXY(0,IER)
                WRITE(*,9000)
  15        CONTINUE
            IR = IST
  20      CONTINUE
```

```
       IR = 20
C
C GET OUTPUT FILENAME AND WRITE IT OUT
C
       KEY = 'ENTER OUTPUT DAMAGE FILENAME : $'
       CALL KEYBD(KEY,R,I,DAMFIL,3,(20),0,IR)
       OPEN(5,FILE=DAMFIL,STATUS='NEW')
       WRITE(5,8000) DAMFIL,DAMTTL,NMODES
       WRITE(5,8010)
       DO 30 I=1,NMODES
         WRITE(5,8020) I,DAM(I),DAMCRT(I)
 30    CONTINUE
 9201 FORMAT(1X,1X,I3\)
 9200 FORMAT(1X,I3\)
 9000 FORMAT(40X\)
 8000 FORMAT('DAMAGE FILENAME : ',A20/A60/'NUMBER OF MODES : ',I2)
 8010 FORMAT('MODE',5X,'D(W)',10X,'Dcrit(W)')
 8020 FORMAT(1X,I2,5X,F7.3,7X,F7.3)
       STOP
       END
```

MODSSUM.FOR

```
$NOFLOATCALLS
      SUBROUTINE MODSSUM
      IMPLICIT INTEGER*2 (I-N)
      DIMENSION ES(100,20)
      CHARACTER*80 KEY
      CHARACTER*40 TITLE
C
C GET CURSOR
C
      CALL GCURXY(IC,IR)
      IR = IR + 1
C
C OPEN UP FILE
C
      READ(5) TITLE,NELM,NMODES
      DO 10 I=1,NMODES
        READ(5) (ES(J,I),J=1,NELM)
 10   CONTINUE
      KEY = 'ENTER WHICH MODE TO SUMMARIZE : $'
      CALL KEYBD(KEY,R,IMODE,A,2,(3),0,IR)
      IR = IR + 1
C
C OPEN UP MODSUM.DD FILE
C
      OPEN(6,FILE='MODSUM.DD',STATUS='NEW')
      WRITE(6,8000) TITLE,NMODES,NELM
 8000 FORMAT(A40,
     1 /'NUMBER OF MODES : ',I2,
     2 /'NUMBER OF ELEMENTS : ',I3)
      WRITE(6,8010) IMODE
 8010 FORMAT(//20X,'SUMMARY FOR MODE NUMBER : ',I2)
      WRITE(6,8020)
 8020 FORMAT(//'ELEMENT',5X,'MODAL STRAIN',
     1    /'  NO.   ',5X,'   ENERGY')
      DO 20 I=1,NELM
        WRITE(6,8030) I,ES(I,IMODE)
 20   CONTINUE
 8030 FORMAT(3X,I3,9X,1PE10.3)
      STOP
      END
```

MODSTRN.FOR

```
$NOFLOATCALLS
C
C PROGRAM MODSTRN - COMPUTE MODAL STRAIN ENERGY MODE BY MODE
C
      IMPLICIT INTEGER*2 (I-N)
      DIMENSION XTRN(200),XAVG(200)
      CHARACTER*80 KEY
      CHARACTER*40 TITLE
      COMMON/XDAT/XTRN,XAVG
C
C CLEAR SCREEN,DISPLAY PROGRAM
C
      KEY = 'MODSTRN - PROGRAM TO COMPUTE MODAL STRAIN ENERGY $'
      CALL VCLEAR
      CALL VCURXY(0,0)
      CALL PRMPT(KEY,1.00)
      CALL GCURXY(IC,IR)
      IR = IR + 1
C
C GET USER INPUTS
C
      KEY = 'ENTER DESCRIPTION OF RUN : $'
      CALL KEYBD(KEY,R,I,TITLE,3,(40),0,IR)
      IR = IR + 1
      KEY = 'ENTER THE NUMBER OF ELEMENTS TO USE : $'
      CALL KEYBD(KEY,R,NELM,A,2,(2),0,IR)
      IR = IR + 1
      KEY = 'ENTER THE NUMBER OF MODES TO USE : $'
      CALL KEYBD(KEY,R,NMODES,A,2,(2),0,IR)
      IR = IR + 1
      CALL VCURXY(0,IR)
C
C OPEN UP BINARY FILE CONTAINING DATA
C
      OPEN(6,FILE='MS.DD',STATUS='NEW',
   1  FORM='UNFORMATTED')
      WRITE(6) TITLE,NELM,NMODES
C
C COMPUTE STRAIN ENERGY FOR 100 PTS FROM 0 TO 1
C
      XLM = 1.0 / NELM
      NELMP1 = NELM + 1
      XPI = 4.0 * ATAN(1.0)
C
C LOOP OVER NUMBER OF MODES
C
      DO 100 IMODE=1,NMODES
      XMX = 0.0
      XCOFF = 1.0
      DO 10 I=1,NELMP1
        XL = FLOAT(I-1) * XLM
        XS = FLOAT(IMODE) * XPI
        XSN = SIN(2*XS*XL)
        XTRN(I) = XCOFF * (XL/2 - XSN/(4.0 * XS))
 10   CONTINUE
C
C COMPUTE ELEMENT STRAIN ENERGY
C
```

```
      DO 15 I=1,NELM
        XAVG(I) = (XTRN(I+1) - XTRN(I))/0.1
        IF(ABS(XAVG(I)).GT.XMX) XMX = XAVG(I)
 15   CONTINUE
C
C SCALE TO UNITY
C
      DO 20 I=1,NELM
        XAVG(I) = XAVG(I) / XMX
 20   CONTINUE
C
C NOW WRITE IT OUT
C
      WRITE(6) (XAVG(I),I=1,NELM)
 100  CONTINUE
      STOP
      END
```

MODSUM.FOR

```
$NOFLOATCALLS
          IMPLICIT INTEGER*2 (I-N)
          DIMENSION ES(100,20),DAM(20),DAMCRT(20)
          CHARACTER*80 KEY
          CHARACTER*60 DAMTTL
          CHARACTER*40 TITLE
          CHARACTER*20 FLNME,DAMFIL
          CHARACTER*3 PDAM(100),ISM
          CHARACTER*27 FMT
          CHARACTER*2 FM
          CHARACTER*1 AES(100,20),FF,FMTM(27),FMM(2)
          DIMENSION ESMX(20),LESMX(20)
          COMMON/DAT/ES,AES,ESMX,LESMX,DAM,DAMCRT,PDAM
          EQUIVALENCE (FMT,FMTM(1)),(FM,FMM(1))
C
C DEFINE FORMFEED
C
          FF = CHAR(12)
          FMT = '(7X,I2,5X, 1(A1,3X),10X,A3)'
          FM = ' 1'
C
C DEISPLAY PROGRAM TO USER
C
          CALL VCLEAR
          CALL VCURXY(0,0)
          KEY = 'MODSUM - PROGRAM TO SUMMARIZE STRAIN ENERGY DATA$'
          CALL PRMPT(KEY,1.00)
          CALL GCURXY(IC,IR)
          IR = IR + 1
C
C GET FILENAME CONTAINING DATA
C
          KEY = 'ENTER FILENAME CONTAINING MODAL STRAIN ENERGIES : $'
          CALL KEYBD(KEY,R,I,FLNME,3,(20),0,IR)
          OPEN(5,FILE=FLNME,FORM='UNFORMATTED')
          IR = IR + 1
C
C SEE IF USER WISHES TO SUMMARIZE MODAL STRIN ENERGIES
C FOR ALL ELEMENTS
C
          KEY='DO YOU WANT TO SUMMARIZE STRAIN ENERGIES FOR A MODE? : $'
          CALL KEYBD(KEY,R,I,ISM,4,(3),0,IR)
          IR = IR + 1
          IF(I.EQ.1) CALL MODSSUM
C
C GET FILENAME CONTAINING DAMAGE INFO
C
          KEY = 'ENTER FILENAME CONTAINING DAMAGE DATA : $'
          CALL KEYBD(KEY,R,I,DAMFIL,3,(20),0,IR)
          IR = IR + 1
          OPEN(7,FILE=DAMFIL)
C
C GET DAMAGE FRACTION
C
          KEY = 'ENTER DAMAGE FRACTION ( <= 1 ) : $'
          CALL KEYBD(KEY,FR,I,A,1,(10),0,IR)
C
C READ IN DATA
```

```
C
          READ(5) TITLE,NELM,NMODES
          WRITE(FM,9111) NMODES
          FMTM(11) = FMM(1)
          FMTM(12) = FMM(2)
 9111  FORMAT(I2)
          READ(7,7010) DAMTTL
 7010  FORMAT(/A60/1X/1X)
          DO 10 I=1,NMODES
            READ(5) (ES(J,I),J=1,NELM)
            READ(7,7020) DAM(I),DAMCRT(I)
 10       CONTINUE
 7020  FORMAT(8X,F7.3,7X,F7.3)
C
C NOW DETERMINE MAX STRAIN ENERGIES FOR EACH MODE
C
          DO 30 I=1,NMODES
            XMX = 0.0
            DO 20 J=1,NELM
                IF(ES(J,I).LE.XMX) GO TO 20
                XMX = ES(J,I)
                LMX = J
 20         CONTINUE
            ESMX(I) = XMX
            LESMX(I) = LMX
 30       CONTINUE
C
C INITIATE AES
C
          DO 35 I=1,NMODES
            DO 34 J=1,NELM
                AES(J,I) = ' '
 34         CONTINUE
 35       CONTINUE
C
C NOW SORT IT ALL OUT
C
C BEGIN BY FINDING ELEMENTS WITH LOW STRAIN ENERGIES
C
          DO 100 I=1,NMODES
            IF(DAM(I).GT.DAMCRT(I)) GO TO 50
            ESMXL = FR * ESMX(I)
            DO 40 J=1,NELM
                IF(ES(J,I).LE.ESMXL) AES(J,I) = 'x'
 40         CONTINUE
            GO TO 100
C
C HERE FOR ELEMENTS WITH HIGH STRAIN ENERGIES
C
 50       CONTINUE
          DO 60 J=1,NELM
            AES(J,I) = 'x'
 60       CONTINUE
 100      CONTINUE
C
C NOW DETERMINE IF ANY ONE ELEMENT MIGHT HAVE DAMAGE
C
          DO 110 IL=1,NELM
```

```
            PDAM(IL) = 'YES'
          DO 120 IM=1,NMODES
              IF(AES(IL,IM).EQ.' ') PDAM(IL) = ' NO'
 120    CONTINUE
 110      CONTINUE
C
C NOW SUMMARIZE DATA
C
          OPEN(6,FILE='MODSUM.DD',STATUS='NEW')
          NPG = (NELM + 39)/40 + 1
          IPAGE = 1
          WRITE(6,9000) TITLE,IPAGE,NPG,DAMFIL,DAMTTL,NMODES,NELM,FR
 9000   FORMAT(//4X,A40,16X,'Page ',I1,' of ',I1,
     1   /4X,'DAMAGE FILE USED : ',A20
     1   /4X,'DAMAGE TITLE : ',A60
     1   /4X,'NUMBER OF MODES : ',I2,
     2   /4X,'NUMBER OF ELEMENTS : ',I2,
     3   /4X,'DAMAGE FRACTION : ',F6.4)
          WRITE(6,9010)
 9010   FORMAT(/4X,'MODE',3X,'MAXIMUM ELEMENT',3X,'ELEMENT',
     1   /4X,' NO.',3X,' STRAIN ENERGY ',3X,'   NO.',/)
          DO 200 I=1,NMODES
            WRITE(6,9020) I,ESMX(I),LESMX(I)
 200      CONTINUE
 9020   FORMAT(5X,I2,6X,1PE10.3,8X,I2)
          WRITE(6,9021) FF
 9021   FORMAT(A1)
          IPAGE = 2
          WRITE(6,9000) TITLE,IPAGE,NPG,DAMFIL,DAMTTL,NMODES,NELM,FR
          WRITE(6,9035)
 9035   FORMAT(//20X,'ELEMENT DAMAGE SUMMARY')
          WRITE(6,9040)
 9040   FORMAT(/4X,'ELEMENT',27X,'MODE',27X,'POSSIBLE',
     1   /4X,'  NO.  ',27X,' NO.',28X,'DAMAGE')
          WRITE(6,9050) (I,I=1,NMODES)
 9050   FORMAT(13X,15(I2,2X))
          DO 70 J=1,NELM
            WRITE(6,FMT) J,(AES(J,I),I=1,NMODES),PDAM(J)
            IF(MOD(J,40).NE.0) GO TO 70
            WRITE(6,9060)
            IPAGE = IPAGE + 1
            WRITE(6,9021) FF
            WRITE(6,9000) TITLE,IPAGE,NPG,DAMFIL,DAMTTL,NMODES,NELM,FR
            WRITE(6,9035)
            WRITE(6,9040)
            WRITE(6,9050) (I,I=1,NMODES)
 70       CONTINUE
          WRITE(6,9060)
 9060   FORMAT(//15X,'x indicates possible modal element damage')
C
C ALL DONE
C
          STOP
          END
```

ANCO

APPENDIX B

SELECTED MATERIAL FROM
MAC/RAN IV USER'S MANUAL

MACRAN IV SYSTEM

Reference Manual

Edited by Paul Agbabian

University Software Systems
250 North Nash Street
El Segundo, California 90245
(213) 640-7616

This manual corresponds to MAC/RAN IV Version 2.0

FOREWORD

The MAC/RAN IV * System is a proven program for the analysis of time series data. Originally developed by Measurement Analysis Corporation of Los Angeles, MAC/RAN was acquired by University Software Systems in 1970. The program has been continuously refined and improved since that time and has served data processors and analysts in a wide variety of applications.

MAC/RAN IV differs from MAC/RAN III in a number of ways. Once an industry standard in the mainframe computer world of the 1970s, the advent of supermini, mini, and microcomputers led to the complete restructuring of MAC/RAN III. MAC/RAN IV can now take full advantage of the greater high speed memory (if available) of modern computers, virtual memory operating systems, and fast mass storage devices. A new, more compact data file structure, along with its direct access allows MAC/RAN to run efficiently on desktop microcomputers as well as the largest of mainframes. The Data File Manager controllers (replacing the EXEC "cards" of MAC/RAN III) reflect this change in data access and structure.

The first three chapters of this manual describe the terms and concepts used in MAC/RAN IV along with the structure of the program. Chapter 4 describes the Data File Manager. Chapters 5 through 15 describe the MAC/RAN IV processors. It is recommended that the user first carefully read Chapters 1 through 4 and study the examples (and possibly the Standard Test Case of Appendix B) to get a feel for the processing sequences. Chapters 5 and up may be referenced as needed.

The PREP (Chapter 5) and PSD (Chapter 8) Processors are particularly powerful and offer many attractive features. PREP contains the digital filters and PSD does spectral averaging (as opposed to simple frequency transformation in FOUR) along with coherency, cross spectra, and transfer function computation. Each computational processor has a section that describes the method or procedures that it uses.

MAC/RAN generally operates in a batch mode and is, in fact, designed to process large streams and many channels (up to 25 at one time) of data efficiently and with minimum user interference. However, if desired, MAC/RAN can now operate on a controller by controller basis, directly from the computer console. The main differences between this mode of operation and a truly interactive one, is that the user must have a grasp of the control language as MAC/RAN does not supply menus or help lists; the reference manual provides the information. The other difference from truly interactive operation is that the plots generated are not immediately displayed, but are written to a file for later postprocessing or output, depending on equipment and installation. Thus MAC/RAN IV remains essentially hardware independent during its operation.

CHAPTER 1 - INTRODUCTION

A. BACKGROUND

1. Time Series Analysis

A series of data values that are compiled with respect to their chronological occurences can be considered a time series. This means that a data value or perhaps an ensemble of values are associated with one instant in time that in most cases precedes and succeedes another pairing or ensemble of values. A simple example might be recording the value of several stocks on a daily basis.

Time series analysis means applying analytic methods to better understand time histories of values or to reduce them into a form where conclusions can more readily be made from the data. Some typical examples of time series analysis might be plotting the data values with respect to their time occurrences, limiting the rates at which these values change (smoothing or filtering), or correlating one set of data points with another.

More often, one is concerned with a transformation of time series data into another domain such that more involved investigation can be facilitated. The most common of these transformations is the Fourier transform which in some sense calculates the frequency content of a time history of values. This transformation is a very powerful tool and is used extensively for many purposes, including power estimation, spectral display and analysis, certain types of correlation, and even mathematical integration and differentiation, among many others. Thus, time series analysis is a very broad term that includes many frequency domain techniques to deal with a wide spectrum of phenomena. These techniques are also applicable to situations that do not involve a time basis, since an analogy can be made with a time series.

2. Solving Time Series Problems

A typical way to solve isolated time series problems is to develop individual, non-integrated programs. This usually requires some research into the techniques employed, a choice of algorithms, and then the time consuming tasks of programming and debugging on a computer. With an arbitrary collection of programs to do the different kinds of signal and data analyses required, problems of compatibility are likely. For example, output from one program may not have the proper format for input to another program. Thus, programs and/or data may have to be partially rewritten nearly every time they are used. Frequent program and data changes increase the probability for human error. Corrections are costly and time-consuming; more significantly, undetected errors may cause distorted or incorrect results.

3. MAC/RAN as a Time Series Tool

MAC/RAN incorporates sophisticated algorithms and data preparation techniques in its processors. This minimizes the signal processing expertise and computer coding necessary to analyze data correctly. All data entered into MAC/RAN remains in a standard data format that allows continuous processing through various modules or selective output at any time during a program run. In addition MAC/RAN processes multiple channels simultaneously for greater throughput.

MAC/RAN allows any one, several, or all of its state-of-the-art processing procedures to be specified within the same program run. The MAC/RAN driver and management software handle the delegation of tasks and data to the necessary modules by way of a command language. Thus, a command file can be easily edited to alter existing sequences of procedures, parameters, or input data. This avoids the rewriting of special software that is required to accommodate any adjustments needed to obtain the desired results.

MAC/RAN is not a programming language nor a set of subroutines from which one constructs his own data reduction system; instead, it is a self-contained data reduction system that provides flexibility in the choice of any specific processing procedures required at any given time.

B. SYSTEM FEATURES

1. Flexibility

As a package of state-of-the-art processing procedures, MAC/RAN performs a wide variety of basic computations to properly reduce time series data. Any one, several, or all of the basic processors (plus the Data File Manager) may be specified within the same program to solve a given problem. Figure 1-1 illustrates the MAC/RAN system.

Processors include:

- Data File Manager (FILIN, FILOUT, MERGE)
- Data Preparation Processor (PREP)
- Fourier Analysis Processor (FOUR)
- Power Spectral Density Processor (PSD)

- Plugboard Simulation Processor (PLUG)
- Display Processor (PLOT)
- Input/Output Port (CTAPE)
- Convolution and Filtering Processor (CONFIL)
- Time and Frequency Analysis Processor (SPEC)
- Amplitude Statistics Processor (AMSTAT)
- Shock Spectrum Analysis Processor (SHOCKS)
- Linear Systems Analysis Processor (TRANS)

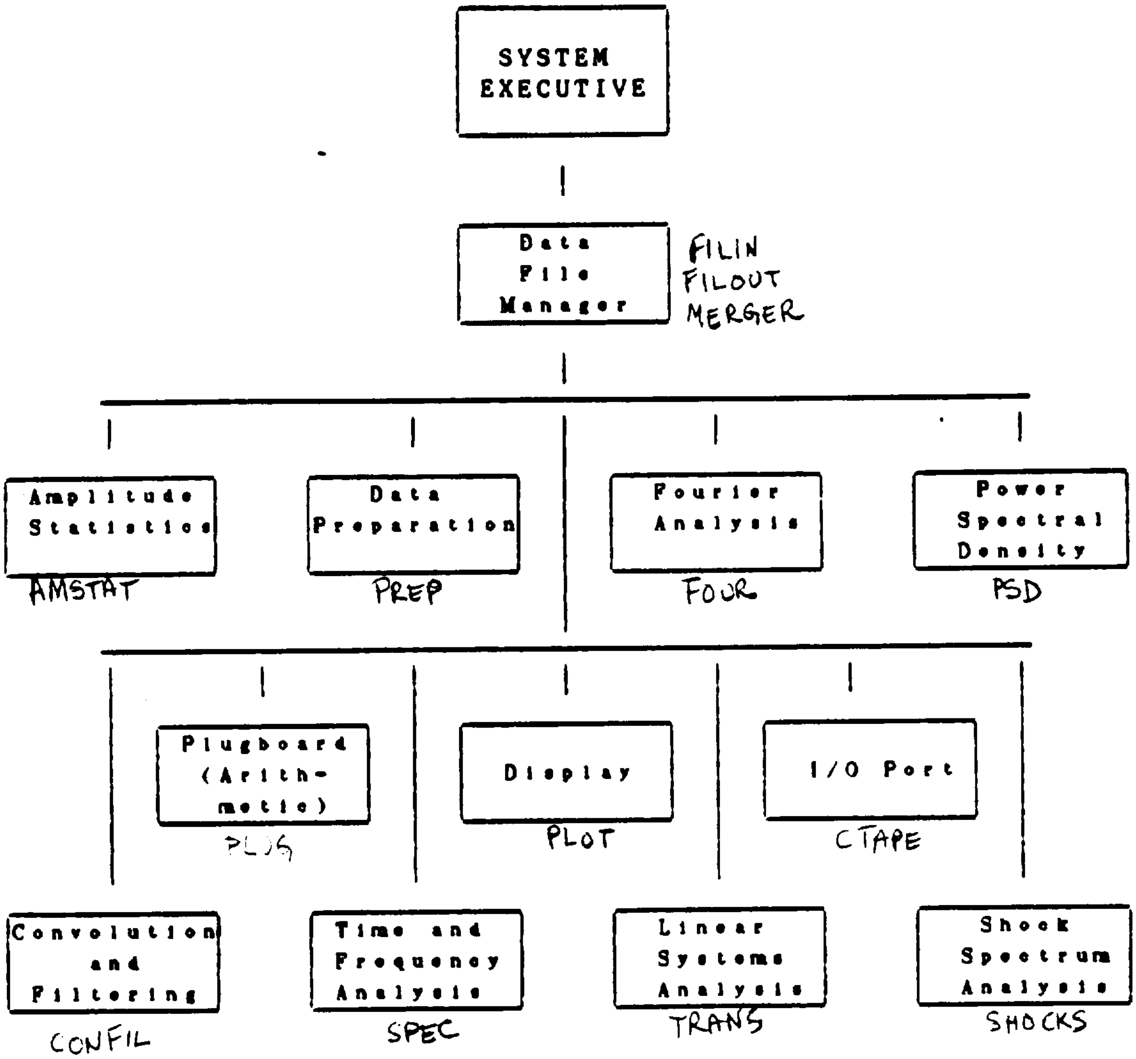


Figure 1-1. MAC/RAN System

2. Simplified Programming Requirements

This modular construction of MAC/RAN provides a simplified approach to processing time series data. The Data File Manager controls the flow of data from processor to processor in order to compute the functions specified in any given run. This feature significantly reduces the amount of control information the user must supply to link each specified processor.

3. Integrated System Capability

Through MAC/RAN's package of general programs, the complete processing for a given data set of time histories can be obtained with a single computer run; in contrast, a non-integrated system would extend the processing over a long period of time with individual computer runs required for each of the individual functions. However, the independence of the processors, even though they can be combined into a single package, allows any given total processor operation to be repeated any number of times in a single run. This flexibility in control of processing can be implemented by the control language parameters.

4. Data Format Compatibility

Throughout the system, every intermediate data format is the same; thus, input to any given processor is in the same format. Consequently, after data is initially fed into the MAC/RAN System, it need never be manually handled again. This greatly simplifies bookkeeping details and increases the efficiency of the overall processing procedure.

5. Display Capability

A special processor is devoted to providing printed and/or plotted outputs of any data or any intermediate functions stored in the standard MAC/RAN format. Inherent within the display processor is the ability to select logarithmic or decimal scales, plus the basic linear scale and scale increment selections in numerically simple values.

C. SYSTEM CONFIGURATION

The MAC/RAN General Process Chart, shown in Figure 1-2, illustrates the flow of data through the system, and the interrelationship between the various stages of processing. These stages are described by symbols representing software, hardware and people.

1. Input

Input to MAC/RAN consists of a control file and data to be analyzed for the desired processors.

2. Control

Control of processing is maintained by the System Manager, invisible to the user, through the parameters in the control file.

3. Output

Generally, a computational processor prints its operational information and the channel statistics of the output data. This processed data is written out as a Standard Data File (SDF). The Display Processor (PLOT, not considered computational) may be used at any time to print or plot data from the SDF of any MAC/RAN processor. Some computational processors optionally print and plot their own output during execution.

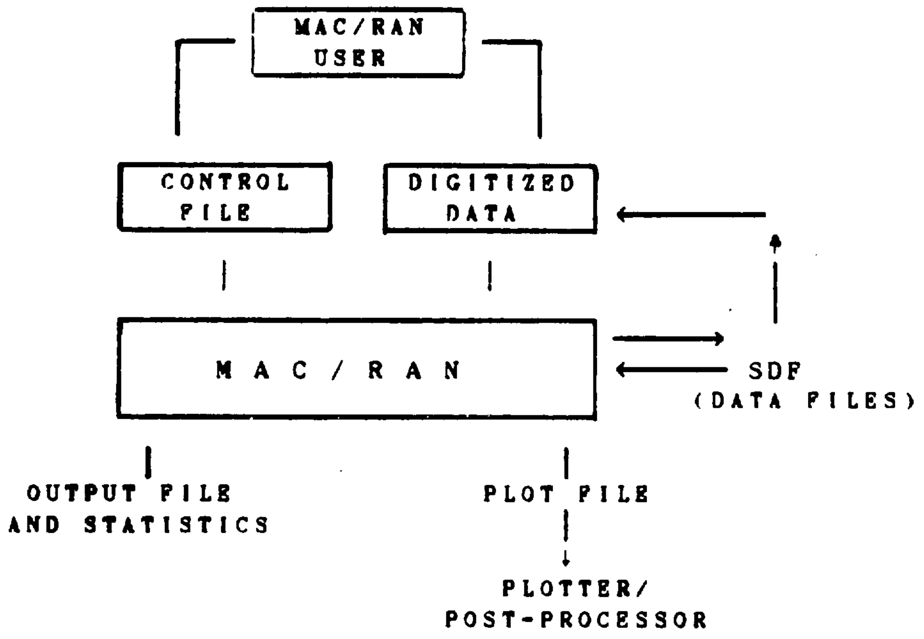


Figure 1-2. MAC/RAN General Process Chart

CHAPTER 2 - SYSTEM OPERATION

A. GENERAL

MAC/RAN consists of a System Executive, a Data File Manager and various processors that do the time series analysis. In addition, there are two special processors that handle I/O and communicate MAC/RAN results to the user, respectively. User inputs to MAC/RAN are a Standard Control File (SCF) and (in most cases) one or more data files. These data files can contain multiple channels of data that are to be processed simultaneously, if desired.

The system operation follows the user commands placed in the input control file. The System Executive recognizes these user specified commands and transfers control to the data file manager or one of the processors.

The MAC/RAN Executive takes care of system initialization, control file information, calls the required processors, and constantly updates the channel statistics during processing. During the course of the processors' computations, the Executive also decides when to write the output data to the appropriate disk file.

The Data File Manager can combine input data files, select or combine channels, and edit the data to be processed. Editing features are start and stop time specification, leading or trailing zero padding, specification of number of points desired, and decimation. Interpolation is possible using the PREP Processor.

B. STANDARD CONTROL FILE (SCF) FORMAT

A Standard Control File consists of various control records, or rows of control information, from which the System Executive drives MAC/RAN. These control records must conform to a standard format of seven fields per record.

The first three fields are alphanumeric and generally contain processor names and processor functions (controllers) as well as annotation information. Field 2 is used only for continuation with certain processors. Fields 4 through 7 are numeric (except when the JBHEAD controller is used). Numbers in these fields can be placed anywhere if a decimal point is supplied, otherwise they must be right justified. Generally, blanks in numeric fields are interpreted as zeros. A complete description of the specific control record formats are given in the following chapters.

The control field format is shown below. The arrows mark the beginning of the field that follows.

Control Field Format:

```
1     78               24          36          48          60          71
↓ f1  ↓↓       f3      ↓     f4    ↓     f5    ↓     f6    ↓     f7
```

Record 1
Record 2
.
.
.

C. MAC/RAN STANDARD DATA FILE (SDF) FORMAT

All data files generated by the system are written in a standard binary format. The first record of a SDF stores channel names and statistics. The subsequent records store channel data. The channel statistics consist of the mean value, variance, maximum value, minimum value, number of channel data points, and sampling rate.

All MAC/RAN processors generate channel statistics when they output a SDF file. The channel (or function) names are defined by the user and they provide annotation for plots. If no names are entered, MAC/RAN supplies them as DATA1, DATA2, etc. If the names are properly chosen, then the plots will be easily identifiable.

D. DEFINITION OF TERMS EMPLOYED

1. General

Controller - A MAC/RAN control word or a part of a control record that specifies control information to the system; e.g., "PLUG" or "FILTLOW" (in the PREP Processor). It must be capitalized.

Control Field - One of seven specific areas of a control record where MAC/RAN control parameters are placed.

Control File - A disk file that contains the control records to drive MAC/RAN in its specific tasks.

Control Group - Adjacent control records with the same controller in a MAC/RAN control file.

Control Record - A record in the MAC/RAN control file that contains a controller and optionally, other control parameters.

DFM - The Data File Manager.

DTSA - "Digital Time Series Analysis" by Otnes & Enochson (Wiley-Interscience, 1972).

Frame - A data group comprising one element of the independent channel and the corresponding function (dependent channel) values that occur simultaneously, e.g. a time slice from a data file that consists of 10 channels. One frame would contain 10 elements--time and nine function values.

Function - The data that resides in a dependent channel.

SCF - A MAC/RAN Standard Control File.

SDF - A MAC/RAN Standard Data File, containing frames of data and channel statistics. Each channel (function) in the SDF has the same length.

2. Mathematical

A time series x_i will be assumed to be sampled at equal spaced intervals separated in time by an amount Δt, so that the continuous form of x_i, $x(t)$, is related to it by

$$x(t) = x(i\Delta t) = x_i \tag{2.1}$$

Without loss of generality, i will be assumed to have the range 0,1,...,N-1 for a total of N values.

The series y_i will frequently be used to represent either a second series, or the original series x_i after some operation such as digital filtering.

Fourier transforms will be denoted with capital letters. For example, the Fourier transform of x(t) will be written as X(f) where f is frequency in Hz. The Fourier transform of x_i will be written X_k where

$$f_k = k\Delta f \tag{2.2}$$

and Δf is the frequency interval in Hz. Usually,

$$f_k = \frac{1}{N\Delta t} \tag{2.3}$$

The sampling rate f_S is usually input to the program rather than Δt, but their relation is quite simple

$$\Delta t = \frac{1}{f_S} \qquad (2.4)$$

The maximum frequency, f_N, sometimes referred to as the Nyquist frequency is

$$f_N = \frac{f_S}{2} = \frac{1}{2\Delta t} \qquad (2.5)$$

CHAPTER 12 - DISPLAY PROCESSOR (PLOT)

A. FUNCTION

The Display Processor provides a convenient means for listing and/or plotting output from any data file in the MAC/RAN SDF format. Any function or combination of functions from a file computed with the MAC/RAN System and written on disk in the SDF format may be saved and later printed or plotted by the Display Processor.

B. INPUT/OUTPUT SPECIFICATIONS

1. Input

a. Control Inputs to the Display Processor include:

- o Printer and/or plot selection
- o Multiple plot selection
- o Plot widths and lengths
- o Plot data intervals
- o Plot scale selection — linear, log, or db

b. Control Inputs to Data File Manager (DFM). The FILIN controller (and optional FILIN CCHAN Controller) is required to describe input disk files.

c. Data Input. Input may consist of one or more channels of time series data in MAC/RAN SDF format.

2. Output

Any desired listing and/or plots with one to 24 curves per plot may be specified. As indicated above, SDFs that have been produced by any computational processor in standard format may be reprinted or replotted at any time.

C. CONTROL RECORDS

Control records for the Display Processor may be input in any order with the exception of multiplot continuation records which must follow the multiplot definition record (see C.1.g below). All control records are optional.

1. Format

The basic format of each record is shown in the following figure.

PLOT

```
1     78               24          36          48          60         71
↓ f1  ↓↓     f3        ↓     f4     ↓     f5     ↓     f6     ↓     f7

PLOT
PLOT   PLOT
PLOT   PRINT
PLOT   PRINTIND
PLOT   INDE    II
PLOT   LOG            T    XCYC        YCYC
PLOT   SCALE   VV     T     XO         XMAX         YO         YMAX
PLOT   NAMEXXXXXXXXXXXX          V1          V2          V3          V4
PLOT   MULTa   II  NN T     XO         XMAX         YO         YMAX
PLOT  CMULT                      V1          V2          V3          V4
PLOT  CNAMEXXXXXXXXXXXX
```

a. <u>Plot/Print Record</u>

Field 1 - PLOT
Field 3 - Print and single plot output options

Blank → Print and plot
PLOT → Plot only - (left adjusted)
PRINT → Print only - (left adjusted)

b. <u>Individual Print Selection Record</u>

Field 1 - PLOT
Field 3 - PRINTIND

If print is requested through Record (a), a simultaneous printout of all input data channels is nominally produced. If the Individual Print Selection Record is input, however, a separated printout of the independent variable along with each dependent variable is produced rather than the simultaneous printout.

c. <u>Independent Variable Selection Record</u>

Field 1 - PLOT
Field 3 - INDE - (left adjusted)
Independent channel number (numerical 1-25) - (columns 16-17, right adjusted)

The independent data channel for printing and for the individual plots is nominally Channel 1. This may be overridden by specifying the independent variable through this controller.

d. Plot Type and Log Plot Cycle Size Selection Record

Field 1 - PLOT
Field 3 - LOG - (left adjusted)

Nominal plot type (numerical 1-8) - (column 23)

1 → Linear x, linear y
2 → Linear x, log y
3 → Log x, linear y
4 → Log x, log y
5 → Linear x, 10 $\log_{10}(y)$
6 → Linear x, 20 $\log_{10}(y)$
7 → Log x, 10 $\log_{10}(y)$ } db
8 → Log x, 20 $\log_{10}(y)$

The nominal plot type (type 1 unless specified by this record) is used for all plots for which no specific plot type is requested (see the SCALE record and MULT record below).

Field 4 - Log cycle size in inches for x-axis (floating point). If not specified, 2.0 is used.

Field 5 - Log cycle size in inches for y-axis. If not specified, 2.0 is used.

e. Scale Selection Record

Field 1 - PLOT
Field 3 - SCALE - (left adjusted)

Dependent channel number, dependent variables only (numerical 2-25). (Columns 16-17, right adjusted.) Defines variable number for which the Scale Selection Record parameters apply. If this field is blank, the record parameters are applied to all individual plots.

Plot Type - (numerical 1-8) - (column 23). Specified plot type is used for plots implied by above channel number field. If blank, nominal plot type is used (see C.1.d above).

Field 4 - Origin for x-axis (floating point)

Field 5 - Maximum value for x-axis (floating point)

If Fields 4 and 5 are not specified, processor selects x-axis scales.

Field 6 - Origin for y-axis (floating point)

Field 7 - Maximum value for y-axis (floating point)

If Fields 6 and 7 are not specified, processor selects y-axis scales.

f. Plot Axis Name Record

Field 1 - PLOT
Field 3 - NAME (left adjusted)
12-character name field (columns 12-23)

Specifies a 12-character label for plot axes for channel numbers given in Fields 4 through 7. If not specified for a channel, the name (ID) field on the data file is used.

Fields 4-7 - Channel numbers to which specified name field applies.

g. Multiple Plot Definition Records

The Multiple Plot Definition Records define a single plot containing multiple dependent variables plotted against a single independent variable on the same set of axes. Only one multiplot may be requested in one execution of the Display Processor.

Field 1 - PLOT
Field 3 - MULT (left adjusted)

Independent Channel Number - (numerical 1-25) - (columns 16-17, right adjusted). Specifies independent variable for multiplot. If omitted, the nominal independent variable is used (see C.1.c).

Number of dependent channels to be plotted - (numerical 1-24) - (columns 20-21, right adjusted). If blank, all data channels except the independent channel are plotted.

Plot Type - (numerical 1-8) - (column 23). If omitted, the nominal plot type is used (see C.1.d).

Field 4 - Origin for x-axis (floating point)

Field 5 - Maximum value for x-axis (floating point)

If Fields 4 and 5 are not specified, processor selects x-axis scales.

Field 6 - Origin for y-axis (floating point)

Field 7 - Maximum value for y-axis (floating point)

If Fields 6 and 7 are not specified, processor selects y-axis scales.

1) Multiplot Continuation Record

Field 1 - PLOT
Field 2 - C
Field 3 - MULT (left adjusted)
Fields 4-7 - Dependent data channels to be plotted - (numerical 1-25) - columns 34-35, 46-47, 58-59, 70-71). If specific dependent data channels are not specified, the first n channels (not including the independent channel) are used where n is specified by the Multiple Plot Definition record above. If fewer than n channels are explicitly requested, the remainder are automatically selected starting from the first variable (omitting the independent channel and avoiding duplication). As many Multiplot Continuation Records as necessary may be input.

2) Multiplot Name Record

Field 1 - PLOT
Field 2 - C
Field 3 - NAME (left adjusted)
12-character label (columns 12-23) for y-axis of multiplot. If not specified, the name field is selected based on the first requested dependent channel.

D. EXAMPLE

```
1     78              24          36          48          60         71
↓ f1  ↓↓      f3       ↓     f4    ↓     f5    ↓     f6    ↓     f7

PLOT
PLOT   INDE     2
PLOT   SCALE   15      4                            10.         1000.
PLOT   LOG             3
PLOT   NAMEEARLY TIMES             2
PLOT   MULT    10   5      10.          53.          6.          900.
PLOT  CMULT                        4           7          12           6
PLOT  CMULT                       21
PLOT  CNAMECOINCIDENCE
```

Explanation:

Record	Effect
1	Produce multiple print and individual plots.
2	Use Channel 2 for independent variable.
3	Use plot type 4 (log-log) on the individual plot of Channel 15. Scale the y-axis on the Channel 15 plot using 10. as the origin and 1000. as the maximum value.
4	Plot all nonspecified plots as log-linear. (In this case, all except Channel 15).
5	Label the Channel 2 axes (in this case, the independent axis) as "EARLY TIMES."
6-8	Produce a multiple plot of the five dependent channel numbers 4, 7, 12, 6, and 21 versus the independent channel number 10. Scale the x-axis using 10. and 53. and y-axis using 6. and 900. as the origins and maximum values. Note: the plot type will be type 3 as specified on Record 4.
9	Label the y-axis on the multiplot as "COINCIDENCE."

CHAPTER 15 - LINEAR SYSTEMS ANALYSIS PROCESSOR (TRANS)

A. FUNCTION

The Linear Systems Analysis Processor (TRANS) operates on spectral density matrices from order 2 up to a maximum of 24. Frequency response functions, coherence functions (ordinary, multiple, and partial) and associated confidence limits are computed for single-input/single-output and for multiple-input/single-output linear systems.

B. INPUT/OUTPUT SPECIFICATIONS

1. Input

a. Control Input to Linear Systems Analysis Processor. These include:

- Specification of output channel
- Plot option for coherence and frequency response functions
- Plot option for confidence limits
- Computation option for complete ordinary coherence matrix

b. Control Inputs to Data File Manager (DFM). FILIN Record required to describe the special spectral density format input file.

c. Data Input. The special spectral density file as generated by either the Time and Frequency Analysis Processor (SPEC) or the Power Spectral Density Processor (PSD).

2. Output

Output from the TRANS Processor includes plots and listings of the frequency response (provided in terms of gain in db and phase in degrees) and coherence functions. Optionally, the resulting transfer functions may be output to SDF.

C. CONTROL RECORDS

TRANS normally requires very little input, as all the information it needs for default execution is on SDF. In summary, the record formats are as follows:

TRANS

Record No.	Field 3	Operation
1	blank	Call TRANS
2	CONF	Set confidence intervals
3	PCON	Plotting of confidence intervals
4	OUT	Defines output function
5	ORDC	Ordinary coherence matrix
6	SCAL	Scaling of plots
7	TAPE	SDF output of the transfer functions

1. Format

a. Record No. 1 (Optional)

Default calling of TRANS. Not required if any other TRANS record is used.

Field 1 - TRANS

b. Record No. 2 (Optional)

Used only if it is desired to change the default confidence limits

Field 1 - TRANS
Field 3 - CONF
Field 4 - Confidence interval for coherence functions

0 → 0.95 (95%) is assumed

Field 5 - Confidence interval for frequency response functions

0 → 0.95 (95%) is assumed

c. Record No. 3 (Optional)

This record causes the confidence intervals to be plotted

Field 1 - TRANS
Field 2 - PCON

d. Record No. 4 (Optional)

Used only if the output function is other than number one.

Field 1 - TRANS
Field 3 - OUT
Field 4 - Output channel number (0 → function number 1)

e. Record No. 5 (Optional)

Used only if it is desired to print and plot the ordinary coherence between all possible input functions in addition to the ordinary coherence between the inputs and the output.

Field 1 - TRANS
Field 3 - ORDC

f. Record No. 6 (Optional, up to 25 may be employed)

Field 1 - TRANS
Field 3 -

Subfield	Columns	Contents
i	8-11	The letters of SCAL.
ii	12-15	GAIN (gain is the only function whose plotting may be varied).
iii	16-17	One or two digits which indicate which data function is to be plotted. For example, GAIN 02 would say that gain two is to be specially plotted. If left blank, all types of functions as defined by subfield ii are scaled according to this record.
iv	18-19	Used only if it is desired to define a second dependent variable, in which case the field contains one or two digits indicating the second function.
v	20-23	A single digit to indicate the plot type: 0,1 Linear-linear 2 Linear x, log y 3 Log x, linear y 4 Log-log 5 Linear x, dB y (20 log y) 7 Log x, dB y (20 log y)

The balance of the special plotting records is as follows:

Field	Columns	Linear	Log
4	24-35	x_{min}	---
5	36-47	x_{max}	number of cycles
6	48-59	y_{min}	---
7	60-71	y_{max}	number of cycles

g. Record No. 7 (Optional)

This is used only if it is desired to output the transfer functions onto SDF.

Field 1 - TRANS
Field 3 - TAPE

If there are more than twelve input functions, only the first twelve transfer functions will be output. Output is in the form of real and imaginary parts. In order for this output to be useful, the "RETAIN" option in PSD will probably have to be requested if PSD was used to set up the input matrix. Otherwise, the number of frequency points will not be commensurate with MAC/RAN FFT lengths. That is, it may not be possible to inverse transform the transfer functions if the "RETAIN" option is not requested.

2. Example

```
1      78           24           36           48           60          71
↓ f1   ↓↓    f3     ↓     f4     ↓     f5     ↓     f6     ↓     f7
```

```
FILIN  Filename
TRANS
END
```

Explanation: This is the default execution of TRANS. Filename is the special spectral density matrix input to TRANS. The standard output will consist of plots and listings of frequency response between the output (default is function no. 1) and all the inputs in addition to the coherence functions. No SDFs of the output transfer functions are generated, hence there is no need for a FILOUT controller. The END controller terminates the run.

D. METHOD

The computational procedures described here will be for computing parameters of a mathematical model assuming a p input $|x_i(t), i = 1,2,...,p|$ and single output $|y(t)|$ linear system. The system parameters to be computed are:

- o Frequency response functions between each of the inputs and the output.
- o Ordinary coherence functions between the inputs and the output, and between all pairs of variables as an option.
- o The multiple coherence function between the output and all of the inputs.
- o Partial (conditional) coherence functions between each input and the output while conditioning on the other inputs.

Chapter 9 of DTSA discusses this problem and lists important references.

1. Computational Procedures

The computational procedures necessary for these options subdivide into three groups:

a. Power and cross-spectral density function computational routines.

b. A procedure for simultaneously handling p + 1 variables to efficiently obtain the spectral density functions among all these variables.

c. The complex variable arithmetic and matrix operations to compute the multidimensional linear system parameters.

The spectral density functions can be computed by the PSD or SPEC Processors. In either case a (p + 1) by (p + 1) set of power and cross-spectral density functions is generated and employed as the input data for this processor.

The first operation performed by this processor is a sorting procedure. The spectral density functions are each supplied as a function of frequency. The Linear Systems Analysis Processor eventually must operate on the (p + 1) by (p + 1) spectral density matrices, one matrix for each frequency value.

The data operated on by the program will be a set of spectral density matrices at frequencies indexed by k as follows:

$$G_{yxx}(k) = \begin{bmatrix} G_{yy}(k) & G_{y1}(k) & G_{y2}(k) & \cdots & G_{yp}(k) \\ G_{1y}(k) & G_{11}(k) & G_{12}(k) & \cdots & G_{1p}(k) \\ \vdots & \vdots & \vdots & & \vdots \\ G_{py}(k) & G_{p1}(k) & G_{p2}(k) & \cdots & G_{pp}(k) \end{bmatrix}$$

$$k = 0,1,\ldots,m \tag{15.1}$$

The frequency index k can represent special frequency values

$$f_k = \frac{kf_C}{m} \qquad k = 0,1,\ldots,m \tag{15.2}$$

where f_C is the Nyquist cut-off frequency. More generally, k can represent the frequency values

$$f_r = f_1 + k\Delta f \qquad k = 0,1,\ldots,m \tag{15.3}$$

where

f_1 = Beginning frequency

Δf = Frequency increment (15.4)

Note: m might denote the maximum lag value if frequency points from (15.2) are used, but might only denote the maximum frequency index if (15.3) is used.

2. Computation of Frequency Response Functions

The mathematical model assumed is indicated in the block diagram of Figure 15-1.

The p input variables and the output variable are assumed to be zero mean, stationary, Gaussian processes whenever any statistical distribution results are discussed. The quantities $H_{iy}(f)$, $i = 1,2,\ldots,p$ are the frequency response function (transfer function) characteristics of the linear systems through which the variables are passing to make up y(t).

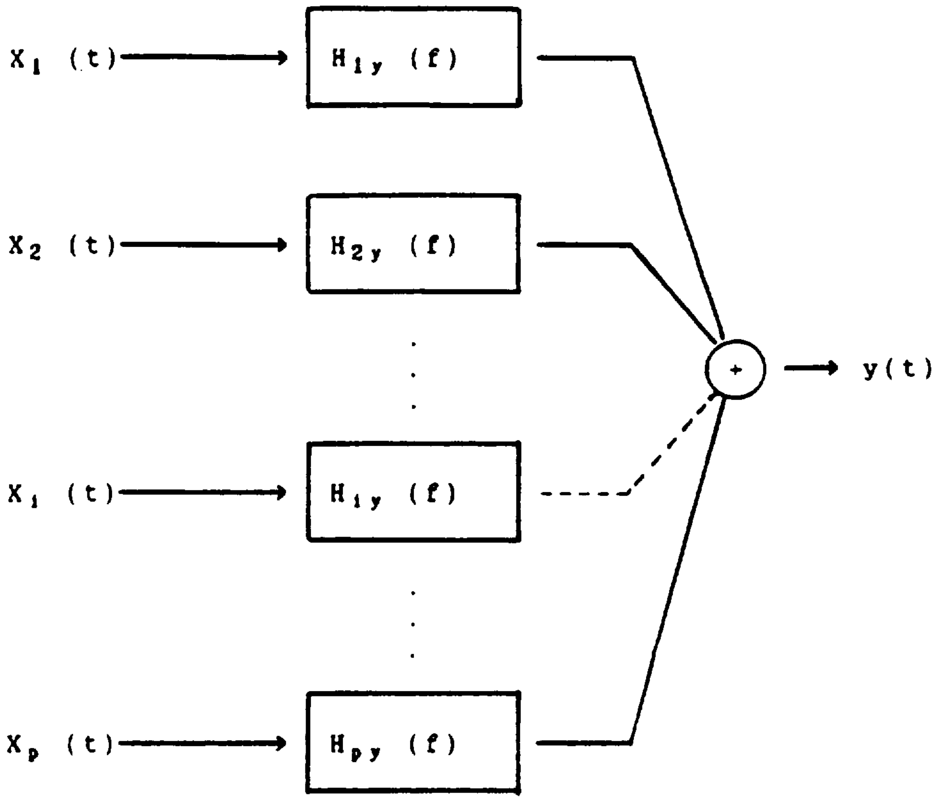


Figure 15-1. Multiple-Input/Single-Output Linear System

The variables $x_i(t)$, $i = 1,2,\ldots,p$ and $y(t)$ are assumed to be discrete (digitized) sequences of N points each. The notation for the N discrete points may be

$$
\begin{aligned}
x_{1n} &= x_1(n\Delta t) \\
x_{2n} &= x_2(n\Delta t) \\
&\vdots \\
x_{pn} &= x_p(n\Delta t) \\
y_n &= y(n\Delta t) \qquad n = 0,1,2,\ldots,N-1 \qquad (15.5)
\end{aligned}
$$

where Δt is the sampling (digitizing) interval.

The matrix equation to be solved to determine the frequency response function is

$$
\begin{bmatrix} G_{1y} \\ G_{2y} \\ \cdot \\ \cdot \\ \cdot \\ G_{py} \end{bmatrix}
=
\begin{bmatrix} G_{11} & G_{12} & \cdots & G_{1p} \\ G_{21} & G_{22} & \cdots & G_{2p} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ G_{p1} & G_{p1} & \cdots & G_{pp} \end{bmatrix}
\begin{bmatrix} H_{1y} \\ H_{2y} \\ \cdot \\ \cdot \\ \cdot \\ H_{py} \end{bmatrix}
\qquad (15.6)
$$

where the function argument has been omitted for notational simplicity (e.g., G_{11} is written instead of $G_{11}(k)$). It is understood that (15.6) is a function of the frequency index k. The matrix and vectors in (15.6) are complex valued and hence require complex arithmetic operations for correct manipulation. In particular,

$$G_{11} = C_{11} - jQ_{11} \qquad (15.7)$$

where $C = C(k)$ and $Q = Q(k)$ are the appropriate co-spectral and quadrature spectral density functions at index value k.

The solution to (15.6) is

$$\begin{bmatrix} H_{1y} \\ H_{2y} \\ \cdot \\ \cdot \\ \cdot \\ H_{py} \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} & \cdots & G_{1p} \\ G_{12} & G_{22} & \cdots & G_{2p} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ G_{p1} & G_{p2} & & G_{pp} \end{bmatrix}^{-1} \begin{bmatrix} G_{1y} \\ G_{2y} \\ \cdot \\ \cdot \\ \cdot \\ G_{py} \end{bmatrix} \qquad (15.8)$$

or in simpler notation

$$H_{xy} = (G_{xx})^{-1} G_{xy} \qquad (15.9)$$

An individual frequency response function is given by

$$H_{iy} = \sum_{l=1}^{p} G^{il} G_{ly} \qquad i = 1,2,\ldots,p \qquad (15.10)$$

3. Computation of Coherence Functions

The ordinary coherence functions between the output y and each input x_i are computed by

$$\gamma^2_{iy} = \frac{|G_{iy}|^2}{G_{ii}\ G_{yy}} = \frac{C^2_{iy} + Q^2_{iy}}{G_{ii}\ G_{yy}} \qquad i = 1,2,\ldots,p \qquad (15.11)$$

The multiple coherence function between the output y and all of the inputs x_1, x_2, ..., x_p is computed by

$$\gamma^2_{y \cdot x} = 1 - [G_{yy}\ G^{yy}]^{-1} \qquad (15.12)$$

where G^{yy} denotes the first diagonal element of the inverse matrix $(G_{xyy})^{-1}$ associated with G_{yxx} of (15.1).

Ordinary and multiple coherence functions for the set of inputs x_i alone are defined by considering the p by p spectral matrix of the inputs.

$$G_{xx} = \begin{bmatrix} G_{11} & G_{12} & \cdots & G_{1p} \\ G_{12} & G_{22} & \cdots & G_{2p} \\ \vdots & \vdots & & \vdots \\ G_{p1} & G_{p2} & \cdots & G_{pp} \end{bmatrix} \qquad (15.13)$$

The ordinary coherence function between any two inputs x_1 and x_1 is computed by

$$\gamma_{11}^{2} = \frac{|G_{11}|^{2}}{G_{11}\ G_{11}} = \frac{C_{11}^{2} + Q_{11}^{2}}{G_{11}\ G_{11}} \qquad (15.14)$$

To obtain the partial coherence function between any input, say x_1, and the output conditioned on the remaining (p - 1) inputs, G_{yxx} is partitioned as indicated below.

$$G_{yxx} = \left[\begin{array}{cc|cccc} G_{yy} & G_{y1} & G_{y2} & \cdots & G_{yp} \\ G_{1y} & G_{11} & G_{12} & \cdots & G_{1p} \\ \hline G_{2y} & G_{21} & G_{22} & \cdots & G_{2p} \\ \vdots & \vdots & \vdots & & \vdots \\ G_{py} & G_{p1} & G_{p2} & \cdots & G_{pp} \end{array}\right] = \begin{bmatrix} \sum_{yy} & \sum_{y1} \\ \sum_{1y} & \sum_{11} \end{bmatrix}$$

(15.15)

Then compute the conditional spectral matrix

$$G_{xy|p} = \sum_{yy} - \sum_{y1} \left(\sum_{11}\right)^{-1} \sum_{1y} \qquad (15.16)$$

This procedure requires the equivalent of the inversion of the (p-1) by (p-1) complex valued matrix Σ_{11}. An individual element $G_{1y|p}$ of the 2x2 matrix $G_{xy|p}$ can be written in terms of real and imaginary parts as

$$G_{iy|p} = C_{iy|p} - jQ_{iy|p} \qquad (15.17)$$

The partial coherence function between the input x_i and the output y, conditioned on the other (p - 1) inputs, is now computed by

$$\gamma^2_{iy|p} = \frac{|G_{iy|p}|^2}{G_{ii|p}\ G_{yy|p}} = \frac{C^2_{iy|p} + Q^2_{iy|p}}{G_{ii|p}\ G_{yy|p}} \qquad (15.18)$$

Similar results apply for x_2 by interchanging x_2 with x_i, for x_3 by interchanging x_3 with x_2, etc.

In the special case of a single-input/single-output linear system, all coherence functions are identical. This can be verified by examining (15.17) for partial coherence and (15.12) for multiple coherence. Upon substituting in values when p = 1, these equations will both reduce to (15.11).

4. Confidence Limit Computations

In addition to the computation of the basic parameter estimates, the confidence limits for the different coherence functions and for the frequency response functions are computed. Confidence limits are supplied for gain and phase of the frequency response functions.

The degrees of freedom, n are given by

$$n = 2B_eT$$

where B_e is the effective spectral resolution bandwidth and T is the effective record length. For the special case of the single-input/single-output system, p = 1 the equations apply to ordinary coherence functions.

For a complete definition and interpretation of the confidence limits, see DTSA.

5. Number of Digits of Precision Plot

Equation (15.18) is not solved explicitly. Rather, a simultaneous equation solution is employed. The computations are performed in a manner that also produces an estimate of the accuracy of results from a numerical viewpoint. In particular, it estimates the number of decimal digits which are significant in the results. As there is one such result for each frequency point, it is essentially a function of frequency for this application. A plot is produced of the digits function, and it is, in fact, the last plot produced by the processor.